ĐẠI HỌC THÁI NGUYÊN
KHOA CÔNG NGHỆ THÔNG TIN

Vũ Vinh Quang – Chủ biên
Nguyễn Đình Dũng, Nguyễn Hiền Trinh, Dương Thị Mai Thương

GIÁO TRÌNH
LÝ THUYẾT THÔNG TIN

THÁI NGUYÊN – NĂM 2010

LỜI MỞ ĐẦU

Giáo trình lý thuyết thông tin được biên soạn dựa trên các bài giảng đã được giảng dạy nhiều năm cho đối tượng là sinh viên chính quy ngành Công nghệ thông tin tại khoa Công nghệ thông tin Đại học Thái Nguyên cùng với việc tham khảo một số giáo trình của các trường Đại học khác cũng như các tài liệu nước ngoài. Để đọc giáo trình này, người đọc cần phải được trang bị đầy đủ các kiến thức về toán cao cấp, xác suất thống kê, lý thuyết thuật toán và một ngôn ngữ lập trình cơ bản (C hoặc Pascal).

Giáo trình được cấu trúc gồm 5 chương

Chương 1 trình bày một số khái niệm cơ bản về lý thuyết thông tin như cấu trúc của hệ thống truyền tin, phân loại môi trường truyền tin, vấn đề rời rạc hóa các nguồn tin liên tục và các khái niệm về điều chế và giải điều chế.

Chương 2 đưa ra các khái niệm cơ bản về tín hiệu và các cơ chế phân tích phổ cho tín hiệu, khái niệm về nhiễu trong quá trình truyền tin.

Chương 3 trình bày các khái niệm cơ bản về độ đo thông tin, lượng tin, entropi và mối quan hệ giữa lượng tin và entropi, các công thức xác định lượng tin và entropi dựa trên cơ sở của lý thuyết xác suất, khái niệm về tốc độ lập tin và thông lượng kênh trong quá trình truyền tin.

Chương 4 giới thiệu các khái niệm chung về mã hóa, điều kiện thiết lập, các phương pháp biểu diễn, các thuật toán mã hóa cơ bản, khái niệm về mã chống nhiễu và mã tuyến tính.

Chương 5 của giáo trình giới thiệu về một số hệ mật mã nổi tiếng trên thế giới để người đọc tham khảo.

Trong quá trình soạn thảo giáo trình chắc chắn không tránh khỏi những thiếu xót về nội dung cũng như hình thức, nhóm biên soạn trân trọng cảm ơn những ý kiến quý báu của các bạn đọc để giáo trình được hoàn thiện hơn.

Thái Nguyên, tháng 01 năm 2010

Thay mặt nhóm biên soạn

Vũ Vinh Quang

CHƯƠNG 1. NHỮNG KHÁI NIỆM CƠ BẢN

1.1 Giới thiệu về lý thuyết thông tin

Trong thế giới ngày nay, chúng ta hàng ngày phải tiếp xúc với rất nhiều các hệ thống chuyển tải thông tin khác nhau như: Các hệ thống truyền hình phát thanh, hệ thống điện thoại cố định và di động, hệ thống mạng LAN, Internet, các hệ thống này đều với mục đích là chuyển thông tin từ nơi phát đến nơi thu với những mục đích khác nhau. Để nghiên cứu về các hệ thống này, chúng ta cần phải nghiên cứu về bản chất thông tin, bản chất của quá trình truyền tin theo quan điểm toán học, cấu trúc vật lý của môi trường truyền tin và các vấn đề liên quan đến tính chất bảo mật, tối ưu hóa quá trình.

Khái niệm đầu tiên cần nghiên cứu là thông tin: thông tin được hiểu là tập hợp các tri thức mà con người thu được qua các con đường tiếp nhận khác nhau, thông tin được mang dưới dạng năng lượng khác nhau gọi là vật mang, vật mang có chứa thông tin gọi là tín hiệu.

Lý thuyết về năng lượng giải quyết tốt vấn đề xây dựng mạch, tín hiệu. Nhưng vấn đề về tốc độ, hiện tượng nhiễu, mối liên hệ giữa các dạng năng lượng khác nhau của thông tin… chưa giải quyết được mà phải cần có một lý thuyết khác đó là lý thuyết thông tin.

Lý thuyết thông tin là lý thuyết nhằm giải quyết vấn đề cơ bản của quá trình truyền tin như vấn đề về rời rạc hóa nguồn, mô hình phân phối xác suất của nguồn và đích, các vấn đề về mã hóa và giải mã, khả năng chống nhiễu của hệ thống...

Cần chú ý rằng lý thuyết thông tin không đi sâu vào việc phân tích các cấu trúc vật lý của hệ thống truyền tin mà chủ yếu nghiên cứu về các mô hình toán học mô tả quá trình truyền tin trên quan điểm của lý thuyết xác suất thống kê, đồng thời nghiên cứu về các nguyên tắc và các thuật toán mã hóa cơ bản, các nguyên tắc mã chống nhiễu...

1.2 Hệ thống truyền tin

Trong thực tế, chúng ta gặp rất nhiều các hệ thống để truyền thông tin từ điểm này tới điểm khác, trong thực tế những hệ thống truyền tin cụ thể mà con

người đã sử dụng và khai thác có rất nhiều dạng, khi phân loại chúng người ta có thể dựa trên nhiều cơ sở khác nhau.

1.2.1 Các quan điểm để phân loại các hệ thống truyền tin

- *Theo năng lượng*
  - Năng lượng một chiều (điện tín)
  - Vô tuyến điện (sóng điện từ)
  - Quang năng (cáp quang)
  - Sóng siêu âm (la-de)
- *Theo biểu hiện bên ngoài*
  - Hệ thống truyền số liệu
  - Hệ thống truyền hình phát thanh
  - Hệ thống thông tin thoại
- *Theo dạng tín hiệu*
  - Hệ thống truyền tin rời rạc
  - Hệ thống truyền tin liên tục

Xuất phát từ các quan điểm đó, trong thực tế trong nhiều lĩnh vực đặc biệt là lĩnh vực truyền thông tồn tại các khái niệm như: Hệ phát thanh truyền hình, hệ truyền tín hiệu số, ...

1.2.2 Sơ đồ truyền tin và một số khái niệm trong hệ thống truyền tin

Định nghĩa: Truyền tin***(transmission)****:* Là quá trình dịch chuyển thông tin từ điểm này sang điểm khác trong một môi trường xác định. Hai điểm này sẽ được gọi là điểm nguồn tin (information source) và điểm nhận tin (information destination). Môi trường truyền tin còn được gọi là kênh tin (chanel).

Sơ đồ khối chức năng của một hệ thống truyền tin tổng quát gồm có 3 thành phần chính: Nguồn tin, kênh tin và nhận tin.

Trong đó:

• ***Nguồn tin:*** là nơi sản sinh ra hay chứa các tin cần truyền đi, hay nguồn tin là tập hợp các tin mà hệ thống truyền tin dùng để tạo các bản tin khác nhau để truyền tin.

• ***Kênh tin****:* là môi trường lan truyền thông tin.

Để có thể lan truyền được thông tin trong một môi trường vật lý xác định, thông tin phải được chuyển thành tín hiệu thích hợp với môi trường truyền lan. Như vậy ta có thể định nghĩa kênh tin:

Kênh tin là nơi hình thành và truyền tín hiệu mang tin đồng thời ở đấy sinh ra các tạp nhiễu phá huỷ thông tin.

Trong lý thuyết truyền tin, kênh là một khái niệm trìu tượng đại diện cho sự hỗn hợp giữa tín hiệu và tạp nhiễu. Từ khái niệm này, sự phân loại kênh sẽ dễ dàng hơn, mặc dù trong thực tế các kênh tin có rất nhiều dạng khác nhau.

Ví dụ:

- Truyền tin theo dây song hành, cáp đồng trục.
- Tín hiệu truyền lan qua các tầng điện ly.
- Tín hiệu truyền lan qua các tầng đối lưu.
- Tín hiệu truyền lan trên mặt đất, trong đất.
- Tín hiệu truyền lan trong nước..

• ***Nhận tin****:* Là cơ cấu khôi phục thông tin ban đầu từ tín hiệu thu được từ đầu ra của kênh

Để tìm hiểu chi tiết hơn ta đi sâu vào các khối chức năng của sơ đồ truyền tin và xét đến nhiệm vụ của từng khối.

1.3 Nguồn tin nguyên thuỷ

1.3.1 Khái niệm chung

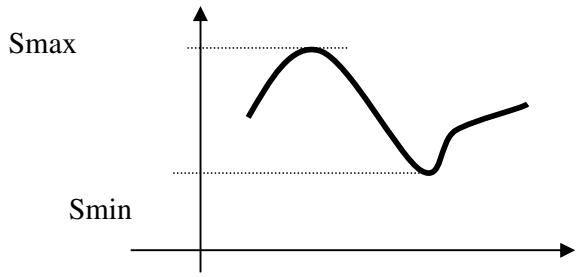
Định nghĩa: Nguồn tin nguyên thuỷ là tập hợp những tin ban đầu mà hệ thống thu nhận được chưa qua một phép biến đối nhân tạo nào.

Về mặt toán học, các tin nguyên thuỷ là những hàm liên tục theo thời gian $f(t)$ hoặc là những hàm biến đổi theo thời gian và một số thông số khác như hình ảnh đen trắng $h(x,y,t)$ trong đó x,y là các toạ độ không gian, hoặc như các thông tin khí tượng $g(\lambda_i,t)$ trong đó λ_i là các thông số khí tượng như nhiệt độ, độ ẩm, tốc độ gió,..

Thông tin nguyên thuỷ cũng có thể là các hệ hàm theo thời gian và các thông số như trường hợp thông tin hình ảnh màu:

$$K(x,y,z)=\begin{cases} f(x,y,z) \\ g(x,y,z). \\ h(x,y,z) \end{cases}$$

Các tin nguyên thuỷ phần lớn là hàm liên tục của thời gian trong một ngưỡng nghĩa là có thể biểu diễn một thông tin nào đó dưới dạng một hàm $S(t)$ tồn tại trong quãng thời gian T và lấy các giá trị bất kỳ trong phạm vi ($S_{\min}, S_{\max}$) trong đó $S_{\min}, S_{\max}$ là ngưỡng nhỏ nhất và lớn nhất mà hệ thống có thể thu nhận được.



Tin nguyên thuỷ có thể trực tiếp đưa vào hệ thống truyền tin nhưng cần phải qua các phép biến đổi sao cho phù hợp với hệ thống tương ứng. Như vậy xét về quan điểm truyền tin thì có hai loại tin và hai loại hệ thống tương ứng:

- Tin rời rạc ứng với
  - Nguồn rời rạc
  - Kênh rời rạc
- Tin liên tục ứng với
  - Nguồn liên tục
  - Kênh liên tục

Sự phân biệt về bản chất của nguồn rời rạc với nguồn liên tục được hiểu là số lượng các tin trong nguồn rời rạc là hữu hạn và số lượng các tin trong nguồn liên tục là không đếm được.

Nói chung các tin rời rạc, hoặc nguyên thuỷ rời rạc, hoặc nguyên thuỷ liên tục đã được rời rạc hoá trước khi đưa vào kênh thông thường đều qua thiết bị mã hoá. Thiết bị mã hoá biến đổi tập tin nguyên thuỷ thành tập hợp những tin thích hợp với đặc điểm cơ bản của kênh như khả năng cho qua (thông lượng), tính chất tín hiệu và tập nhiễu.

1.3.2 Bản chất của thông tin theo quan điểm truyền tin

Chỉ có quá trình ngẫu nhiên mới tạo ra thông tin. Một hàm gọi là ngẫu nhiên nếu với một giá trị bất kì của đối số, giá trị của hàm là một đại lượng ngẫu nhiên (các đại lượng vật lí trong thiên nhiên như nhiệt độ môi trường, áp suất không khí… là hàm ngẫu nhiên của thời gian).

Một quá trình ngẫu nhiên được quan sát bằng một tập các giá trị ngẫu nhiên. Quá trình ngẫu nhiên được coi là biết rõ khi thu nhận và xử lí được một tập đủ nhiều các giá trị đặc trưng của nó.

Giả sử quá trình ngẫu nhiên X(t) có một tập các giá trị mẫu (hay còn được gọi là các biến) $x(t)$, khi đó ta biểu diễn quá trình ngẫu nhiên bởi:

$$X(t) = \{x(t)\}_{x \in X}$$

Ví dụ: Quan sát thời gian vào mạng của các sinh viên trong 1 ngày, người ta tiến hành phỏng vấn 10 sinh viên, gọi X là thời gian vào mạng, x_k là thời gian vào mạng của sinh viên thứ $k, (k = 1,2,...,10)$ ta thu được mẫu như sau:

$X = \{10, 50, 20, 150, 180, 30, 30, 5, 60, 0\}$ đơn vị tính (phút)

Việc đoán trước một giá trị ngẫu nhiên là khó khăn. Ta chỉ có thể tìm được quy luật phân bố của các biến thông qua việc áp dụng các qui luật của toán thống kê để xử lý các giá trị của các biến ngẫu nhiên mà ta thu được từ các tín hiệu.

Quá trình ngẫu nhiên có thể là các hàm trong không gian 1 chiều, khi đó ta có quy luật phân phối xác suất 1 chiều và hàm mật độ phân phối xác suất được xác định bởi các công thức

$$F(x) = p(X < x); \quad w(x) = \frac{dF(x)}{dx}$$

Trong đó:

- x là biến ngẫu nhiên
- p(x) xác suất xuất hiện $X = x$ trong quá trình ngẫu nhiên, thường được viết là $p(x) = p(X = x)$.

Nếu quá trình ngẫu nhiên là các hàm trong không gian 2 chiều khi đó quy luật ngẫu nhiên được biểu hiện bởi các công thức

$$F(x, y) = p(X < x; Y < y);\ w_{xy}(x, y) = \frac{\partial^2 F}{\partial x \partial y}.$$

Tương tự, ta cũng có các quy luật phân phối xác suất trong không gian nhiều chiều.

Các đặc trưng quan trọng của biến ngẫu nhiên

1. Trị trung bình (kì vọng toán học) của một quá trình ngẫu nhiên $X(t)$

$$E(X) = \overline{X(t)} = \int_{-\infty}^{+\infty} x(t)w(x)dx$$

2. Trị trung bình bình phương

$$E^2(X) = \overline{X^2(t)} = \int_{-\infty}^{+\infty} x^2(t)w(x)dx$$

3. Phương sai

$$D(X) = \overline{\left(X - \overline{X}\right)^2} = \int_{-\infty}^{+\infty} \left(x(t) - E(x)\right)^2 w(x)dx$$

4. Hàm tương quan

Mô tả mối quan hệ thống kê giữa các giá trị của 1 quá trình ngẫu nhiên ở các thời điểm t_1, t_2

$$B_x(t_1, t_2) = E\left(X(t_1), X(t_2)\right) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1 x_2 w(x(t_1), x(t_2))dx_1 dx_2$$

Nếu hai quá trình X, Y khác nhau ở hai thời điểm khác nhau, khi đó

$$B_{xy}(t_1, t_2) = E\left(X(t_1), Y(t_2)\right) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xyw(x(t_1), y(t_2))dxdy$$

Để giải quyết bài toán một cách thực tế, người ta không thể xác định tức thời mà thường lấy trị trung bình của quá trình ngẫu nhiên. Có hai loại trị trung bình theo tập hợp và trị trung bình theo thời gian. Ta cần nghiên cứu trị trung bình theo tập hợp, tuy vậy sẽ gặp nhiều khó khăn khi tiếp nhận và xử lý tức thời các biến ngẫu nhiên vì các biến ngẫu nhiên luôn biến đổi theo thời gian. Để tính trị trung bình theo thời gian, ta chọn thời gian đủ lớn để quan sát các biến ngẫu nhiên dễ ràng hơn vì có điều kiện quan sát và sử dụng các công thức thống kê, khi đó việc tính các giá trị trung bình theo thời gian được xác định bởi các công thức:

$$\overline{X(t)} = \underset{T\to+\infty}{Lim}\frac{1}{T}\int_0^T x(t)dt$$

Trị trung bình bình phương:

$$\overline{X^2(t)} = \underset{T\to+\infty}{Lim}\frac{1}{T}\int_0^T x^2(t)dt$$

Khi thời gian quan sát T dần đến vô cùng thì trị trung bình tập hợp bằng trị trung bình thời gian. Trong thực tế ta thường chọn thời gian quan sát đủ lớn chứ không phải vô cùng như vậy vẫn thoả mãn các điều kiện cần nhưng đơn giản hơn, khi đó ta có trị trung bình theo tập hợp bằng trị trung bình theo thời gian. Ta có:

$$\overline{X(t)} = \int_{-\infty}^{+\infty} x(t)w(x)dx = \underset{T\to+\infty}{Lim}\frac{1}{T}\int_0^T x(t)dt$$

Tương tự:

$$\overline{X^2(t)} = \int_{-\infty}^{+\infty} x^2(t)w(x)dx = \underset{T\to+\infty}{Lim}\frac{1}{T}\int_0^T x^2(t)dt$$

Trường hợp này gọi chung là quá trình ngẫu nhiên dừng theo hai nghĩa:

- Theo nghĩa hẹp: Trị trung bình chỉ phụ thuộc khoảng thời gian quan sát $\tau = t_2 - t_1$ mà không phụ thuộc gốc thời gian quan sát.

• Theo nghĩa rộng: Gọi là quá trình ngẫu nhiên dừng khi trị trung bình là một hằng số và hàm tương quan chỉ phụ thuộc vào hiệu hai thời gian quan sát $\tau = t_2 - t_1$. Khi đó ta có mối tương quan

$$B_x(t_1,t_2) = B(\tau = t_2 - t_1) = B(\tau) = \overline{X(t).X(t+\tau)}$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1 x_2 w(x_1,x_2)dx_1dx_2 = \lim_{T\to+\infty}\frac{1}{T}\int_{-\infty}^{T} x(t)x(t+\tau)dt$$

Tóm lai: Để nghiên cứu định lượng nguồn tin, hệ thống truyền tin mô hình hoá nguồn tin bằng 4 quá trình sau:

1. Quá trình ngẫu nhiên liên tục: Nguồn tiếng nói, âm nhạc, hình ảnh là tiêu biểu cho quá trình này.

2. Quá trình ngẫu nhiên rời rạc: là quá trình ngẫu nhiên liên tục sau khi được rời rạc hóa theo mức trở thành quá trình ngẫu nhiên rời rạc.

3. Dãy ngẫu nhiên liên tục: Đây là trường hợp một nguồn liên tục đã được gián đoạn hóa theo thời gian, như thường gặp trong các hệ thống tin xung như: điều biên xung, điều tần xung ... không bị lượng tử hóa.

4. Dãy ngẫu nhiên rời rạc: Nguồn liên tục được gián đoạn theo thời gian hoặc trong hệ thống thông tin có xung lượng tử hoá.

1.4 Hệ thống kênh tin

1.4.1 Khái niệm

Như chúng ta đã biết: vật chất chỉ có thể dịch chuyển từ điểm này đến một điểm khác trong một môi trường thích hợp và dưới tác động của một lực thích hợp. Trong quá trình dịch chuyển của một dòng vật chất, những thông tin về nó hay chứa trong nó sẽ được dịch chuyển theo. Đây chính là bản chất của sự lan truyền thông tin.

Vậy có thể nói rằng việc truyền tin chính là sự dịch chuyển của dòng các hạt vật chất mang tin (tín hiệu) trong môi trường. Trong quá trình truyền tin, hệ thống truyền tin phải gắn được thông tin lên các dòng vật chất tạo thành tín hiệu và lan truyền đi.

Việc tín hiệu lan truyền trong một môi trường xác định chính là dòng các hạt vật chất chịu tác động của lực, lan truyền trong một cấu trúc xác định của

môi trường. Dòng vật chất mang tin này ngoài tác động để dịch chuyển, còn chịu tác động của các lực không mong muốn trong cũng như ngoài môi trường. Đây cũng chính là nguyên nhân làm biến đổi dòng vật chất không mong muốn hay là nguyên nhân gây ra nhiễu trong quá trình truyền tin.

Như vậy: Kênh tin là môi trường hình thành và truyền lan tín hiệu mang tin đồng thời ở đó sinh ra các tạp nhiễu phá hủy thông tin.

1.4.2 Phân loại môi trường truyền tin

Kênh tin là môi trường hình thành và truyền lan tín hiệu mang tin. Để mô tả về kênh chúng ta phải xác định được những đặc điểm chung, cơ bản để có thể tổng quát hoá về kênh.

Khi tín hiệu đi qua môi trường do tác động của tạp nhiễu trong môi trường sẽ làm biến đổi năng lượng, dạng của tín hiệu. Mỗi môi trường có một dạng tạp nhiễu khác nhau. Vậy ta có thể lấy sự phân tích, phân loại tạp nhiễu để phân tích, phân loại cho môi trường (kênh)

- Môi trường trong đó tác động nhiễu cộng là chủ yếu $N_c(t)$:

Nhiễu cộng là nhiễu sinh ra một tín hiệu ngẫu nhiên không mong muốn và tác động cộng thêm vào tín hiệu ở đầu ra. Nhiễu cộng là do các nguồn nhiễu công nghiệp sinh ra, luôn luôn tồn tại trong các môi trường truyền lan tín hiệu.

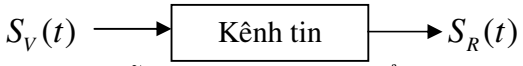
- Môi trường trong đó tác động nhiễu nhân là chủ yếu $N_n(t)$:

Nhiễu nhân là nhiễu có tác động nhân vào tín hiệu, nhiễu này gây ra do phương thức truyền lan của tín hiệu, hay là sự thay đổi thông số vật lý của bộ phận môi trường truyền lan khi tín hiệu đi qua. Nó làm nhanh, chậm tín hiệu (thường ở sóng ngắn) làm tăng giảm biên độ tín hiệu.

- Môi trường gồm cả nhiễu cộng và nhiễu nhân

1.4.3 Mô tả sự truyền tin qua kênh:

Xét hệ thống truyền tin trong đó $S_V(t)$ là thông tin truyền, $S_R(t)$ là thông tin thu



Ta có biểu thức mô tả nhiễu trong trường hợp tổng quát

$$S_R(t) = N_n(t)S_V(t) + N_c(t)$$

Trong thực tế, ngoài các nhiễu cộng và nhiễu nhân, tín hiệu cũng chịu tác động của hệ số đặc tính xung của kênh $H(t)$ do đó:

$$S_R(t) = N_n(t).H(t).S_V(t) + N_c(t)$$

Đặc tính kênh không lý tưởng này sẽ gây ra một sự biến dạng của tín hiệu ra so với tín hiệu vào gọi là méo tín hiệu và là một nguồn nhiễu trong quá trình truyền tin.

Tín hiệu vào của kênh truyền hiện nay là những dao động cao tần với những thông số biến đổi theo quy luật của thông tin. Các thông số có thể là biên độ, tần số hoặc góc pha, dao động có thể liên tục hoặc gián đoạn, nếu là gián đoạn sẽ có những dãy xung cao tần với các thông số xung thay đổi theo thông tin như biên độ xung, tần số lặp lại, thời điểm xuất hiện. Trong trường hợp dao động liên tục, biểu thức tổng quát của tín hiệu có dạng sau:

$$S_V(t) = a(t)\cos(\omega t + \beta(t))$$

trong đó $a(t)$ là biên độ, ω: tần số góc, $\beta(t)$: góc pha, các thông số này biến đổi theo quy luật của thông tin để mang tin và nhiễu tác động sẽ làm thay đổi các thông số này làm sai lệch thông tin trong quá trình truyền.

Theo mô hình mạng của kênh tin, kí hiệu $p(y/x)$ là xác suất nhận được tin $y(t)$ khi đã phát đi tin $x(t)$, nếu đầu vào ta đưa vào tin $x(t)$ với xác suất xuất hiện là $p(x)$ ta nhận được ở đầu ra một tin $y(t)$ với xác suất xuất hiện $p(y)$ ứng với $x(t)$. Với yêu cầu truyền tin chính xác, ta cần phải đảm bảo $y(t)$ phải là tin nhận được từ $x(t)$ tức là $p(y/x) = 1$. Điều này chỉ có được khi kênh không có nhiễu. Khi kênh có nhiễu, có thể trên đầu ra của kênh chúng ta nhận được một tin khác với tin được phát, có nghĩa là $p(y/x) < 1$ và nếu nhiễu càng lớn thì xác suất này càng nhỏ. Như vậy về mặt toán học, chúng ta có thể sử dụng xác suất $p(y/x)$ là một tham số đặc trưng cho đặc tính truyền tin của kênh.

1.5 Hệ thống nhận tin

Nhận tin là đầu cuối của hệ thống truyền tin. Nhận tin thường gồm có bộ nhận biết thông tin và xử lý thông tin. Nếu bộ phận xử lý thông tin là thiết bị tự động ta có một hệ thống truyền tin tự động.

Vì tín hiệu nhận được ở đầu ra của kênh là một hỗn hợp tín hiệu và tạp nhiễu xảy ra trong kênh, nên nói chung tín hiệu ra không giống với tín hiệu đưa vào kênh. Nhiệm vụ chính cần thực hiện tại nhận tin là từ tín hiệu nhận được $y(t)$ phải xác định được $x(t)$ nào được đưa vào ở đầu vào của kênh. Bài toán này được gọi là bài toán thu hay phục hồi tín hiệu tại điểm thu.

1.6 Một số vấn đề cơ bản của hệ thống truyền tin

Các vấn đề lý thuyết thông tin cần giải quyết trong quá trình truyền tin là: hiệu suất, độ chính xác của quá trình truyền tin trong đó.

1.6.1 Hiệu suất (tốc độ lập tin)

Là lượng thông tin nguồn lập được trong một đơn vị thời gian với độ sai sót cho phép.

1.6.2 Độ chính xác (hay khả năng chống nhiễu của hệ thống)

Là khả năng giảm tối đa sai nhầm thông tin trên đường truyền, yêu cầu tối đa với bất kỳ một hệ thống truyền tin nào là thực hiện được sự truyền tin nhanh chóng và chính xác. Những khái niệm về lý thuyết thông tin cho biết giới hạn tốc độ truyền tin trong một kênh tin, nghĩa là khối lượng thông tin lớn nhất mà kênh cho truyền qua với một độ sai nhầm nhỏ tùy ý.

Trong nhiều trường hợp nguồn tin nguyên thủy là liên tục nhưng dùng kênh rời rạc để truyền tin. Vậy nguồn tin liên tục trước khi mã hóa phải được rời rạc hóa. Để xác minh phép biến đổi nguồn liên tục thành nguồn rời rạc là một phép biến đổi tương đương 1–1 về mặt thông tin, trước hết ta khảo sát cơ sở lý thuyết của phép rời rạc hóa gồm các định lý lấy mẫu và quy luật lượng tử hóa.

1.7 Rời rạc hóa một nguồn tin liên tục

Trong các hệ thống truyền tin mà thiết bị đầu và cuối là những thiết bị xử lý thông tin rời rạc như các hệ thống truyền số liệu thì không cho phép truyền trực tiếp tin liên tục. Do vậy nếu các nguồn tin là liên tục, nhất thiết trước khi đưa tin vào kênh phải thông qua một phép biến đổi liên tục thành rời rạc. Sau đó sẽ áp dụng các phương pháp mã hóa để đáp ứng được các chỉ tiêu kỹ thuật của hệ thống truyền tin cụ thể.

Phép biến đổi nguồn tin liên tục thành rời rạc gồm hai quá trình cơ bản:

- Quá trình rời rạc hóa theo thời gian hay là khâu lấy mẫu.
- Quá trình lượng tử hóa.

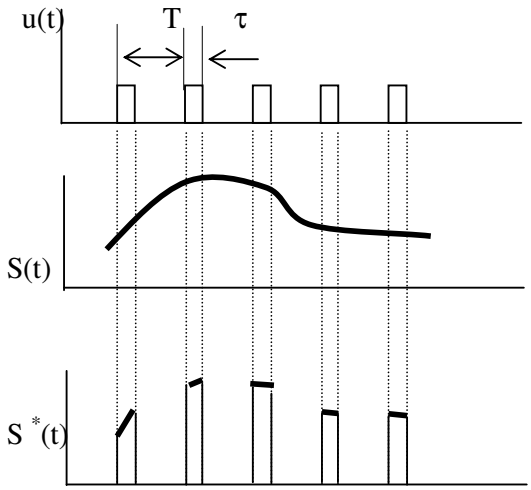
Cơ sở lý thuyết của phép biến đổi này gồm các định lý lấy mẫu và luật lượng tử hóa như sau.

1.7.1 Quá trình lấy mẫu

Giả sử nguồn tin liên tục dạng tín hiệu được biểu diễn bằng hàm tin phụ thuộc thời gian $S(t) = a(t)\cos(\omega t + \beta)$

Việc lấy mẫu một hàm tin có nghĩa là trích từ hàm đó ra các mẫu tại những thời điểm nhất định. Nói một cách khác là thay hàm tin liên tục bằng một hàm rời rạc là những mẫu của hàm trên lấy tại những thời điểm gián đoạn. Vấn đề đặt ra ở đây là xét các điều kiện để cho sự thay thế đó là một sự thay thế tương đương. Tương đương ở đây là về ý nghĩa thông tin, nghĩa là hàm thay thế không bị mất mát thông tin so với hàm được thay thế.

Việc lấy mẫu có thể thực hiện bằng một rơ le điện, điện tử bất kì đóng mở dưới tác động của điện áp $u(t)$ nào đó. Thời gian đóng mạch của rơ le là thời gian lấy mẫu τ, chu kỳ lấy mẫu là T, tần suất lấy mẫu là $f = \frac{1}{T}$. Từ $S(t)$ liên tục, ta thu được $S^*(t)$ theo nghĩa rời rạc (Hình 1.1)



Hình 1.1

Trong kỹ thuật, việc lấy mẫu phải thỏa mãn một số điều kiện của định lý lấy mẫu trong không gian thời gian cho quá trình ngẫu nhiên có băng tần hạn chế.

Sau đây chúng ta xét một số khái niệm

• Biến đổi Fourier: hàm $S(t)$ được gọi là có biến đổi Fourier là $S(f)$ nếu: $S(f) = \int_{-\infty}^{+\infty} S(t)e^{-j2\pi ft}dt$

• Giả sử có tín hiệu liên tục $S(t) = \int_{-\infty}^{+\infty} S(f)e^{-j2\pi ft}df$ có biến đổi Fourier là $S(f)$ được gọi là có băng tần hạn chế nếu $S(f) = 0$ với $|f| > f_{\max}$, trong đó $f_{\max}$ là tần số cao nhất của tín hiệu $S(t)$. Một tín hiệu như thế được biểu diễn một cách duy nhất bởi những mẫu của $S(t)$ với tần số lấy mẫu là f_S với $f_S \geq 2f_{\max}$. Ta thấy ngoài miền tần số $(-f_{\max}, f_{\max})$ năng lượng coi như bằng 0 nên:

$$S(t) = \int_{-f_{\max}}^{+f_{\max}} S(f)e^{-j2\pi ft}df$$

• Tín hiệu có băng tần hạn chế được lấy mẫu với tần số lấy mẫu là $f_S = 2f_{\max}$ có thể khôi phục lại từ các mẫu của nó theo công thức nội suy sau:

$$S(t) = \sum_{n=-\infty}^{n=+\infty} S\left(\frac{n}{2f_{\max}}\right) \frac{\sin\left[2\pi f_{\max}\left(t - \frac{n}{2f_{\max}}\right)\right]}{2\pi f_{\max}\left(t - \frac{n}{2f_{\max}}\right)}$$

trong đó: $\left\{S\left(\frac{n}{2f_{\max}}\right)\right\}$ là các mẫu của $S(t)$ lấy tại $t = \frac{n}{2f_{\max}}$ với $n = 0, \pm 1, \pm 2, ...$

Như vậy nếu thời gian lấy mẫu đủ dài và số mẫu đủ lớn thì năng lượng của tín hiệu lấy mẫu tương đương với năng lượng của tín hiệu gốc. Các kết quả trên được phát biểu bởi định lý sau đây:

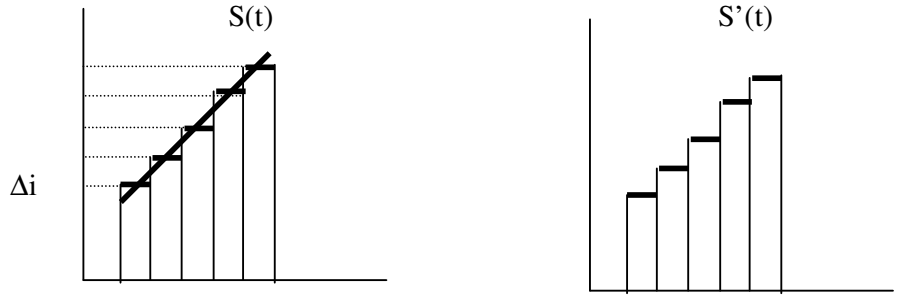
Định lý lấy mẫu Shanon*:* Hàm $S(t)$ trong khoảng $(-f_{\max}, f_{\max})$ hoàn toàn được xác định bằng cách lấy mẫu với tần số lấy mẫu $f_S = 2f_{\max}$.

1.7.2 Khâu lượng tử hoá

Giả thiết hàm tin $S(t)$ biến thiên liên tục với biên độ của nó thay đổi trong khoảng $(S_{\min}, S_{\max})$. Ta chia khoảng $(S_{\min}, S_{\max})$ thành n khoảng:

$$S_{\min} = S_0 < S_1 < S_2 < ... < S_n = S_{\max}$$

Như vậy hàm tin liên tục $S(t)$ qua phương pháp rời rạc sẽ biến đổi thành $S'(t)$ có dạng biến đổi bậc thang gọi là hàm lượng tử hoá với mỗi mức lượng tử $\Delta_i = S_{i+1} - S_i, (i = 0..n-1)$. Sự lựa chọn các mức Δ_i thích hợp sẽ giảm sự sai khác giữa $S(t)$ và $S'(t)$.



Hình 1.2

Phép biến đổi $S(t)$ thành $S'(t)$ được gọi là phép lượng tử hoá. $\Delta_i, (i = 0,...,n-1)$ gọi là mức lượng tử hoá.

Nếu $\Delta_i = \dfrac{S_{\max} - S_{\min}}{n}, \forall i = 0,...,n-1$, ta có qui luật lượng tử hoá đều ngược lại ta gọi là lượng tử hóa không đồng đều. Do sự biến thiên $S(t)$ trong thực tế thường là không đều nên người ta thường dùng qui luật lượng tử không

đều. Việc chia lưới lượng tử không đều này phụ thuộc vào mật độ xác suất các giá trị tức thời của $S(t)$. Ta thường chọn Δ_i sao cho các giá trị tức thời của $S(t)$ trong phạm vi Δ_i là hằng số. Về mặt thống kê, phép lượng tử hóa chính là việc tạo mẫu phân khoảng với độ dài khoảng là Δ_i và ứng với mỗi khoảng xác định tần số xuất hiện của tín hiệu trong khoảng, khi đó ta nhận được bảng phân khoảng của tín hiệu tương ứng sau khi đã rời rạc hóa.

Tóm lại: Việc biến một nguồn liên tục thành một nguồn rời rạc cần hai phép biến đổi: lấy mẫu và lượng tử hoá. Thứ tự thực hiện hai phép biến đổi này phụ thuộc vào điều kiện cụ thể của hệ thống:

- Lượng tử hoá sau đó lấy mẫu.
- Lấy mẫu sau đó lượng tử hoá.
- Thực hiện đồng thời hai phép biến đổi trên.

1.8 Điều chế và giải điều chế

1.8.1 Điều chế

Trong các hệ thống truyền tin liên tục, các tin hình thành từ nguồn tin liên tục được biến đổi thành các đại lượng điện (áp, dòng) và chuyển vào kênh. Khi muốn chuyển các tin ấy qua một cự ly lớn, phải cho qua một phép biến đổi khác gọi là điều chế.

Định nghĩa: Điều chế là phép biến đổi nhằm chuyển thông tin ban đầu thành một dạng năng lượng thích hợp với môi trường truyền lan sao cho năng lượng ít bị tổn hao, ít bị nhiễu trên đường truyền tin.

Các phương pháp điều chế:

Các phương pháp điều chế cao tần thường dùng với tín hiệu liên tục

- Điều chế biên độ AM (Amplitude Modulation)
- Điều chế đơn biên SSB (Single Side Bande)
- Điều tần FM (Frequency Modulation)
- Điều pha PM (Phase Modulation)

Với tín hiệu rời rạc, các phương pháp điều chế cao tần cũng giống như trường hợp thông tin liên tục, nhưng làm việc gián đoạn theo thời gian gọi là manip hay khóa dịch. Gồm các phương pháp sau.

- Manip biên độ ASK (Amplitude Shift Key)

- Manip tần số FSK (Frequency Shift Key)
- Manip pha PSK (Phase Shift Key)

1.8.2 Giải điều chế

Định nghĩa: Giải điều chế là nhiệm vụ thu nhận, lọc, tách thông tin nhận được dưới dạng một điện áp liên tục hay một dãy xung điện rời rạc giống như đầu vào với một sai số cho phép.

Các phương pháp giải điều chế

Về phương pháp giải điều chế nói cách khác là phép lọc tin, tùy theo hỗn hợp tín hiệu nhiễu và các chỉ tiêu tối ưu về sai số (độ chính xác) phải đạt được mà chúng ta có các phương pháp lọc tin thông thường như:

- Tách sóng biên độ,
- Tách sóng tần số
- Tách sóng pha

CHƯƠNG 2. TÍN HIỆU

2.1 Một số khái niệm cơ bản

Tín hiệu là các thông tin mà con người thu nhận được từ môi trường bên ngoài thông qua các giác quan hay các hệ thống đo lường. Ví dụ như: Sóng địa chấn, nhịp tim của bệnh nhân, lưu lượng của các dòng chảy hay âm thanh, sóng điện từ, tín hiệu số,…. Về mặt toán học, tín hiệu được hiểu như một hàm số phụ thuộc vào thời gian $S(t)$. Sau đây chúng ta sẽ nghiên cứu các dạng tín hiệu cơ bản.

2.1.1 Tín hiệu duy trì

Thể hiện sự duy trì của tín hiệu với cường độ không thay đổi theo thời gian, tín hiệu được biểu hiện bằng hàm số

$$I(t) = \begin{cases} a, & t \geq 0, \\ 0, & t < 0 \end{cases} \tag{2.1}$$

trong đó a là cường độ của tín hiệu. Tín hiệu duy trì thể loại tín hiệu không thay đổi trong suốt quãng thời gian, ví dụ tiếng ù của âm thanh, nhịp phát manip với giá trị không đổi, ánh sáng với cùng một cường độ,…

2.1.2 Tín hiệu xung

Biểu hiện tín hiệu xuất hiện đột ngột trong khoảng thời gian cực nhỏ với cường độ cực kỳ lớn sau đó không xuất hiện

$$\partial(t) = \begin{cases} +\infty, & t = 0, \\ 0, & t \neq 0. \end{cases} \tag{2.2}$$

Tín hiệu xung thường rất hay gặp trong các tín hiệu đo của các thiết bị vật lý hay cơ học.

2.1.3 Tín hiệu điều hoà

Biểu hiện các loại tín hiệu tuần hoàn trong một khoảng chu kì nào đó, được biểu diễn bằng công thức tổng quát

$$S(t) = A\cos(\omega t + \beta) \tag{2.3}$$

Trong đó: A là biên độ dao động, $f = \frac{\omega}{2\pi}$ là tần số, $T = \frac{2\pi}{\omega}$ là chu kỳ của dao động cơ bản. Dao động cơ bản còn có thể biểu diễn bằng công thức tổng quát hơn

$$S(t) = a\cos\omega t + b\sin\omega t \qquad (2.4)$$

Khi đó ta có thể biểu diễn dao động cơ bản như một vectơ trong hệ trục tọa độ cực hay dưới dạng số phức tổng quát $S(t) = re^{j\omega t}$ với j là đơn vị ảo.

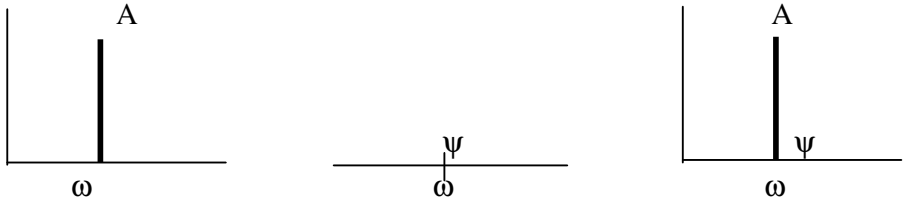
2.2 Phân tích phổ cho tín hiệu

Trong thực tế, một tín hiệu ngẫu nhiên gồm hữu hạn hay vô hạn các tín hiệu đơn sắc (nguyên tố), khi đó để nghiên cứu và xử lý tín hiệu ngẫu nhiên bất kỳ, chúng ta phải tìm cách tách từ tín hiệu ngẫu nhiên thành từng tín hiệu đơn sắc, việc phân tích đó gọi là phép phân tích phổ.

Nếu tín hiệu điều hoà có dạng:

$$S(t) = A\cos(\omega t + \psi),$$

khi đó chúng ta có các khái niệm phổ biên độ, phổ pha và phổ thực như sau:



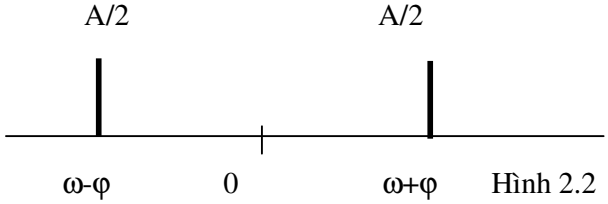
Phổ biên độ Phổ pha Phổ thực

Hình 2.1

Trong các loại phổ trên, năng lượng tập trung chủ yếu ở ω.

Nếu tín hiệu cho dưới dạng phức $S(t) = \frac{A}{2}\left(e^{(j\omega t+\varphi)} + e^{(j\omega t-\varphi)}\right)$

Khi đó chúng ta có dạng phổ phức



Hình 2.2

2.2.1 Chuỗi Fourier và phổ rời rạc

Định nghĩa 1

Cho 2 hàm số $\varphi(x),\psi(x)$ liên tục khả tích trên $[a,b]$, định nghĩa

$$<\varphi,\psi>=\int_a^b \varphi(x)\psi(x)dx \tag{2.5}$$

được gọi là tích vô hướng của 2 hàm trên không gian $C_{[a,b]}$. Kí hiệu

$$\|\varphi\|_{[a,b]}=\sqrt{\int_a^b \varphi^2(x)dx} \tag{2.6}$$

được gọi là chuẩn của $\varphi(x)$ trên $C_{[a,b]}$.

Định nghĩa 2

Cho hệ hàm $\varphi_1(x),\varphi_2(x),...,\varphi_n(x),....$ xác định liên tục trên $[a,b]$.

Hệ $\{\varphi_k(x)\}_1^\infty$ được gọi là hệ trực giao nếu thỏa mãn điều kiện

$$<\varphi_k,\varphi_l>=\begin{cases} 0 & ,k\neq l \\ \|\varphi_k\|^2 & ,k=l. \end{cases} \tag{2.7}$$

Hệ $\{\varphi_k(x)\}_1^\infty$ được gọi là hệ trực chuẩn nếu thỏa mãn điều kiện

$$<\varphi_k,\varphi_l>=\begin{cases} 0 & ,k\neq l \\ 1 & ,k=l. \end{cases} \tag{2.8}$$

Nhận xét: Với mọi hệ trực giao bất kỳ, luôn luôn tồn tại phép biến đổi về hệ trực chuẩn bằng

$$\varphi_k(x):=\frac{\varphi_k(x)}{\|\varphi_k\|}. \tag{2.9}$$

Định nghĩa 3

Cho hệ $\{\varphi_k(x)\}_1^\infty$ là một hệ trực giao và $f(x)$ là một hàm số bất kỳ xác định liên tục trên $[a,b]$, khi đó khai triển

$$f(x)=\sum_{k=1}^{+\infty}A_k\varphi_k(x) \tag{2.10}$$

được gọi là khai triển Fourier tổng quát thông qua hệ trực giao trong đó A_k được gọi là hệ số khai triển.

Để xác định các hệ số khai triển, ta nhân hai vế với $\varphi_n(x)$ và lấy tích phân trên đoạn $[a,b]$, ta được

$$\int_a^b f(x)\varphi_n(x)dx=\sum_{k=1}^{+\infty}A_k\int_a^b\varphi_k(x)\varphi_n(x)dx$$

Do tính chất trực giao của hệ $\{\varphi_k(x)\}_1^{\infty}$ ta thu được

$$\int_a^b f(x)\varphi_n(x)dx=\sum_{k=1}^{+\infty}A_k\int_a^b\varphi_k(x)\varphi_n(x)dx=A_n\int_a^b\varphi_n^2(x)dx$$

Hay $$\langle f,\varphi_n\rangle=A_n\|\varphi_n\|^2$$

Tức là

$$A_n=\frac{\langle f,\varphi_n\rangle}{\|\varphi_n\|^2}=\frac{\int_a^b f(x)\varphi_n(x)dx}{\int_a^b\varphi_n^2(x)dx}. \tag{2.11}$$

Công thức (2.7) là công thức xác định hệ số khai triển Fourier trong trường hợp tổng quát với một hệ trực giao bất kỳ.

Sau đây ta xét một số ví dụ áp dụng phương pháp khai triển với các hệ trực giao khác nhau

Ví dụ 1: Xét hệ $\{\sin kx\}_1^{+\infty}$ trên đoạn $[0,2\pi]$

Ta có

$$\int_0^{2\pi}\sin kx\sin lxdx=\frac{1}{2}\int_0^{2\pi}\big(\cos(k-l)x-\cos(k+l)x\big)dx=0,$$

$$\int_0^{2\pi} \sin^2 kx dx = \frac{1}{2}\int_0^{2\pi} (1-\cos 2kx) dx = \pi.$$

Tức là

$$\langle \sin kx, \sin lx \rangle = \begin{cases} 0 & , \quad k \neq l, \\ \pi & , \quad k = l. \end{cases}$$

Hay nói cách khác, hệ $\{\sin kx\}_1^{+\infty}$ là trực giao trên đoạn $[0, 2\pi]$. Khi đó xét hàm $f(x)$ bất kỳ, ta luôn có khai triển $f(x) = \sum_{k=1}^{+\infty} A_k \sin kx$ trong đó

$$A_k = \frac{1}{\pi}\int_0^{2\pi} f(x) \sin kx dx.$$

Hoàn toàn tương tự, ta cũng chứng minh được các hệ hàm $\{\sin kx\}_1^{+\infty}$, $\{\sin kx, \cos lx\}_1^{+\infty}$ là các hệ trực giao trên các đoạn tương ứng.

Tổng quát, có thể chứng minh rằng hệ $\varphi_k(x) = \sin \frac{2k\pi x}{T}, (k = 1, 2, ...)$ trực giao trên đoạn $\left[-\frac{T}{2}, \frac{T}{2}\right]$.

Ví dụ 2: Giả sử quan sát tín hiệu $S(t)$ tuần hoàn với chu kỳ T trong khoảng thời gian $\left(-\frac{T}{2}, \frac{T}{2}\right)$, xét hệ hàm

$$\varphi_k(t) = e^{jk\frac{2\pi}{T}t}, k = 0, \pm 1, \pm 2, ...$$

Ta có thể chứng minh rằng hệ hàm $\{\varphi_k(t)\}$ trực giao trên đoạn $\left(-\frac{T}{2}, \frac{T}{2}\right)$ tức là

$$\left\langle \varphi_k, \varphi_{-l} \right\rangle = \int_{-\frac{T}{2}}^{\frac{T}{2}} \varphi_k(t)\varphi_{-l}(t)dt = \begin{cases} 0, & k \neq l, \\ T, & k = l. \end{cases}$$

Khi đó sử dụng phương pháp khai triển Fourier, ta khai triển hàm $S(t)$ thông qua hệ hàm trực giao

$$S(t) = \sum_{k=-\infty}^{+\infty} A_k \varphi_k(t)$$

Trong đó:

$$A_0 = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)dt,\ A_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)e^{-j\frac{2k\pi}{T}t}dt.$$

Hay

$$S(t) = A_0 + \sum_{k=1}^{+\infty}\left(A_k e^{j\frac{2k\pi}{T}t} + A_{-k}e^{-j\frac{2k\pi}{T}t} \right)$$

$$= A_0 + \sum_{k=1}^{+\infty}\left(A_k\left(\cos\frac{2k\pi t}{T} + j\sin\frac{2k\pi t}{T} \right) + A_{-k}\left(\cos\frac{2k\pi t}{T} - j\sin\frac{2k\pi t}{T} \right) \right)$$

$$= A_0 + \sum_{k=1}^{+\infty}\left(A_k + A_{-k} \right)\cos\frac{2k\pi t}{T} + j\left(A_k - A_{-k} \right)\sin\frac{2k\pi t}{T}$$

$$= A_0 + \sum_{k=1}^{+\infty} a_k \cos\frac{2k\pi t}{T} + jb_k \sin\frac{2k\pi t}{T}$$

Trong đó

$$A_0 = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)dt,\ a_k = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)\cos\frac{2k\pi t}{T}dt,\ b_k = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)\sin\frac{2k\pi t}{T}dt$$

Trong trường hợp tín hiệu là hàm số chẵn tức là hàm số $S(t)\sin\frac{2k\pi t}{T}$ là hàm số lẻ, khi đó hệ số $b_k \equiv 0, \forall k = 1,2,3,...$ Khi đó

$$S(t) = A_0 + \sum_{k=1}^{+\infty} a_k \cos\frac{2k\pi t}{T}.$$

Hoàn toàn tương tự, nếu tín hiệu là hàm số lẻ tức là hàm số $S(t)\cos\frac{2k\pi t}{T}$ là hàm số lẻ, khi đó hệ số $a_k \equiv 0, \forall k = 0,1,2,3,...$ Khi đó

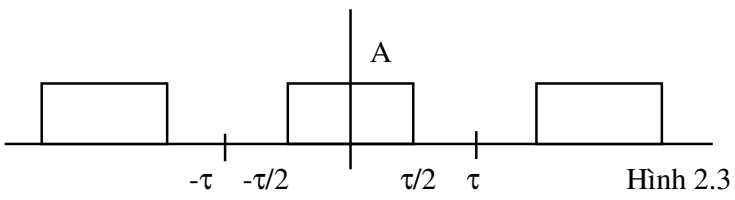
$$S(t) = \sum_{k=1}^{+\infty} b_k \sin\frac{2k\pi t}{T}.$$

Nhận xét:

+ Với một tín hiệu tuần hoàn với chu kỳ T thì hệ hàm $\varphi_k(t) = e^{jk\frac{2\pi}{T}t}, k = 0,\pm1,\pm2,...$ được chọn là hệ trực giao tổng quát trên đoạn $\left[-\frac{T}{2},\frac{T}{2}\right]$ trong đó nếu tín hiệu là chẵn thì hệ trực giao được xác định là $\left\{\cos\frac{2k\pi}{T}t\right\}_0^{+\infty}$ còn nếu tín hiệu là lẻ thì hệ trực giao được xác định là $\left\{\sin\frac{2k\pi}{T}t\right\}_0^{+\infty}$

+ Đối với một tín hiệu bất kỳ thì chúng ta cần phải xác định chu kỳ của tín hiệu cũng như tính chất chẵn lẻ của tín hiệu trước khi khai triển.

Ví dụ 3: Phân tích phổ cho tín hiệu là dãy xung sau:



Hình 2.3

Ta có chu kỳ của tín hiệu là $T = 2\tau$. Xét trên đoạn $\left[-\frac{T}{2}, \frac{T}{2}\right]$, khi đó

$$S(t) = \begin{cases} A, & t \in \left[-\frac{\tau}{2}, \frac{\tau}{2}\right], \\ 0, & t \notin \left[-\frac{\tau}{2}, \frac{\tau}{2}\right]. \end{cases}$$

Tín hiệu $S(t)$ là hàm chẵn. Sử dụng các công thức khai triển với hệ trực giao $\left\{\cos\frac{2k\pi}{T}t\right\}_0^{+\infty}$ ta có $S(t) = A_0 + \sum_{k=1}^{+\infty} A_k \cos\frac{k\pi}{\tau}t$ trong đó

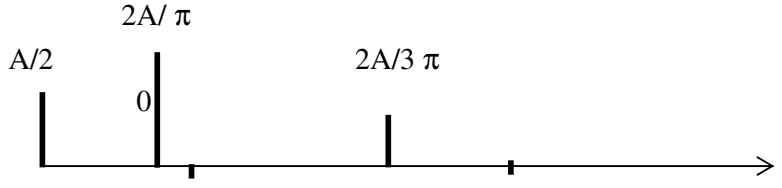
$$A_0 = \frac{1}{2\tau}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)dt = \frac{1}{2\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} Adt = \frac{A}{2}.$$

$$A_k = \frac{2}{2\tau}\int_{-\frac{T}{2}}^{\frac{T}{2}} S(t)\cos\frac{k\pi}{\tau}tdt = \frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} A\cos\frac{k\pi}{\tau}tdt = \frac{2A}{k\pi}\sin\frac{k\pi}{2}$$

$$\text{Hay } A_k = \begin{cases} \frac{2A}{k\pi}(-1)^l, & k = (2l+1), \\ 0, & k = 2l. \end{cases}$$

Như vậy ta có khai triển

$$S(t) = \frac{A}{2} + \sum_{k=1}^{+\infty} \frac{2A}{(2k+1)\pi}(-1)^k \cos\frac{(2k+1)\pi}{\tau}t.$$



Hình 2.4

Phổ của tín hiệu được mô tả bởi hình 2.4

2.2.2 Tích phân Fourier và phổ liên tục

Với tín hiệu liên tục ta có hàm $S(t)$ trong phổ thời gian tương ứng với $S(j\omega)$ trong phổ tần số. Sử dụng công thức khai triển Fourier trong trường hợp tổng quát, ta có:

$$S(j\omega) = f\left[S(t)\right] = \int_{-\infty}^{+\infty} S(t)e^{-j\omega t}dt \tag{2.12}$$

Ngược lại ta có:

$$S(t) = f\left[S(j\omega)\right] = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S(j\omega)e^{j\omega t}d\omega \tag{2.13}$$

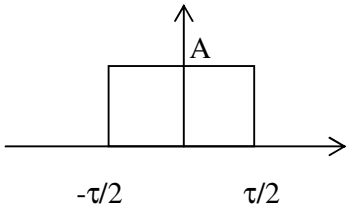
Tưng tự như xét với $S(t)$ ta có phổ của $S(j\omega)$ như sau

- Phổ phức: $S(j\omega) = A(\omega) + jB(\omega)$.
- Phổ biên độ: $= \sqrt{A^2(\omega) + B^2(\omega)}$.
- Phổ pha: $= Arctg\left[\frac{B(\omega)}{A(\omega)}\right]$.

Ví dụ:

Xét một xung vuông sau:

$$S(j\omega) = \int_{-\infty}^{+\infty} S(t)e^{-j\omega t}dt$$

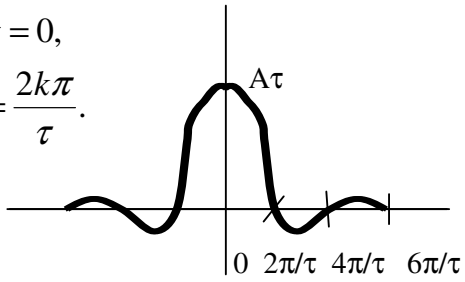


Hình 2.5

Ta có:

$$S(j\omega) = \int_{-\infty}^{+\infty} S(t)e^{-j\omega t}dt = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} Ae^{-j\omega t}dt = \frac{A}{-j\omega}\left(e^{-j\frac{\tau\omega}{2}} - e^{j\frac{\tau\omega}{2}}\right) = A\tau\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}$$

Như vậy phổ $S(j\omega) = \begin{cases} A\tau, & t = 0, \\ 0, & t = \dfrac{2k\pi}{\tau}. \end{cases}$



Hình 2.6

2.2.3 Phổ các tín hiệu điều chế

Tín hiệu thông tin muốn truyền đi xa phải nhờ tín hiệu cao tần. Để tín hiệu cao tần mang thông tin ta phải làm cho tín hiệu cao tần biến thiên theo qui luật của tín hiệu thông tin. Tín hiệu cao tần có dạng:

$$S(t) = a_0 \cos(\omega_0 t + \beta) = a_0 \cos\psi(t) \tag{2.14}$$

Ta có thể điều chế 2 thông số biên độ a_0 và góc $\psi(t)$. Với góc $\psi(t)$ ta có thể điều chế theo tần số ω_0 (gọi là tín hiệu điều tần) theo góc pha β (gọi là điều pha). Sau đây chúng ta sẽ xét chi tiết các phương pháp điều chế.

Phương pháp điều biên

Trong phương pháp điều biên, ta biến đổi biên độ của tín hiệu cao tần theo qui luật của thông tin $u(t)$ tức là biến đổi có chứa lượng tin cần truyền, còn tần số và góc pha không đổi.

Giả sử tin cần truyền là $u(t)$, khi đó ta có công thức biến đổi:

$$S(t) = [a_0 + M_0 u(t)]\cos(\omega_0 t + \beta) \tag{2.15}$$

trong đó: $M_0 = \dfrac{\Delta_a}{a_0}$ được gọi là hệ số điều chế, trong kỹ thuật điều chế, để thông tin điều chế đảm bảo độ chính xác, ta cần chọn $M_0 \le 1$. Hàm số $u(t)$ được gọi là hàm tin, hàm tin thường chọn là hàm đơn sắc, nếu hàm tin là các thông tin phức tạp, ta phải tách thành các tín hiệu đơn sắc bằng phương pháp phân tích phổ đã nghiên cứu ở chương trước.

Giả sử $u(t)$ là hàm đơn sắc có dạng một dao động điều hoà

$$u(t) = \cos(\Omega t + \theta)$$

Khi đó ta có

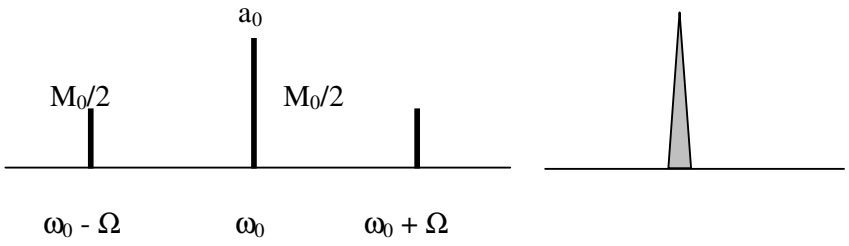
$$S(t) = [a_0 + M_0 \cos(\Omega t + \theta)]\cos(\omega_0 t + \beta)$$

$$= a_0 \cos(\omega_0 t + \beta) + M_0 \cos(\Omega t + \theta)\cos(\omega_0 t + \beta)$$

$$= a_0 \cos(\omega_0 t + \beta) + \frac{M_0}{2}\cos[(\omega_0 + \Omega)t + \theta + \beta] + \frac{M_0}{2}\cos[(\omega_0 - \Omega)t + \beta - \theta]$$

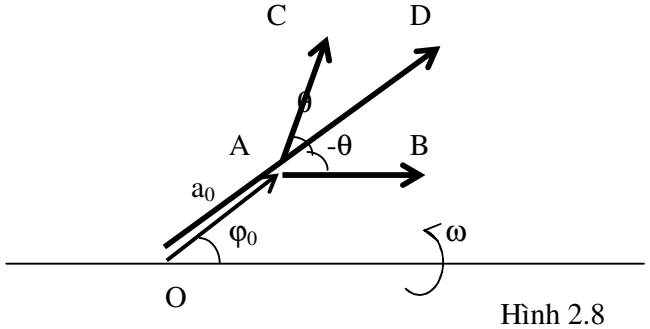
$$= S_1(t) + S_2(t) + S_3(t)$$

Như vậy tín hiệu qua quá trình điều biên sẽ gồm ba thành phần, Một thành phần $S_1(t)$ dao động với tần số mang ω_0 và 2 thành phần $S_2(t), S_3(t)$ dao động với tần số biên $(\omega_0 \pm \Omega)$. Biên độ của tần số biên bằng nhau và bằng $\frac{M_0}{2}$.



Hình 2.7

Trong trường hợp tín hiệu là không đơn sắc thì tín hiệu điều biên là một miền, không phải là phổ vạch, đồ thị véc tơ của tín hiệu điều biên như sau



Hình 2.8

Trong đó

OA: Tín hiệu mang

AB, AC: Tần số biên

OD: Tín hiệu điều chế

Nhận xét:

• OD max khi AD=AB + AC = $Ma_0 \Rightarrow OD = a_0 + Ma_0 = a_0(1+M)$. Để không nhiễu thì $AD \le a_0$ hay $M \le 1$.

• OD // OA: Thì tín hiệu hàm tin là đơn sắc và phổ là phổ vạch nếu hàm tin không đơn sắc thì phổ là một miền.

• Theo đồ thị thì biên độ sóng mang (a_0) lớn chiếm nhiều hơn 70% năng lượng nên thường nén để tiết kiệm năng lượng giảm hao phí.

Phương pháp điều tần

Trong phương pháp điều tần, người ta biến đổi tần số của sóng mang theo tín hiệu của hàm tin $u(t)$, tức là

$$\omega_0 := \omega_0 + \Delta_\omega u(t).$$

Trong đó hệ số Δ_ω gọi là hệ số điều tần.

Xuất phát từ công thức tích phân

$$\psi(t) = \omega_0 t + \beta = \int \omega_0 dt$$

Qua quá trình điều tần, ta nhận được

$$\psi(t) = \int [\omega_0 + \Delta_\omega u(t)]dt = \omega_0 t + \Delta_\omega \int u(t)dt$$

Giả sử sóng mang có dạng $S(t) = a_0 \cos(\omega_0 t + \beta)$ và hàm tin là hàm đơn sắc $u(t) = \cos(\Omega t + \theta)$. Khi đó qua phương pháp điều biên

$$S(t) = a_0 \cos\left(\omega_0 t + \Delta_\omega \int u(t)dt\right) = a_0 \cos\left(\omega_0 t + \Delta_\omega \int \cos(\Omega t + \theta)dt\right)$$

$$= a_0 \cos\left(\omega_0 t + \frac{\Delta_\omega}{\Omega}\sin(\Omega t + \theta)\right) = a_0 \cos\left(\psi_1(t) + \psi_2(t)\right)$$

Như vậy qua quá trình điều tần, pha của sóng mang đã được tách thành 2 thành phần $\psi_1(t)$ chứa tần số của sóng mang và thành phần $\psi_2(t)$ chứa thành phần tin $u(t)$.

Phương pháp điều pha

Tương tự như phương pháp điều tần, phương pháp điều pha biến đổi góc pha có chứa hàm tin $u(t)$ còn biên độ và tần số không đổi. Ta có công thức biến đổi trong trường hợp tín hiệu đơn sắc:

$$\psi(t) = \omega_0 t + \beta + \Delta_\beta u(t) = \omega_0 t + \beta + \Delta_\beta \cos(\Omega t + \theta)$$

Tức là

$$S(t) = a_0 \cos\left(\omega_0 t + \beta + \Delta_\beta \cos(\Omega t + \theta)\right)$$

Nhận xét: Về hình thức thì có thể coi tín hiệu điều tần, điều pha giống nhau trong công thức tổng quát sau đây:

$$S(t) = a_0 \cos\left(\omega_0 t + \beta_1 + \Delta_m \cos(\Omega t + \theta_1)\right) \tag{2.16}$$

Tín hiệu điều tần thì $\Delta_m = \dfrac{\Delta_\omega}{\Omega}, \beta_1 = \beta, \theta_1 = \theta - \dfrac{\pi}{2}$, tín hiệu điều pha thì

$$\Delta_m = \Delta_\beta, \beta_1 = \beta, \theta_1 = \theta .$$

2.2.4 Phân tích tín hiệu ngẫu nhiên

Do các tín hiệu ngẫu nhiên là các đại lương ngẫu nhiên tuân theo các quy luật phân phối xác định nên việc phân tích các tín hiệu ngẫu nhiên dựa trên cơ sở phân tích mối tương quan giữa các đại lượng ngẫu nhiên của lý thuyết xác suất thống kê.

Phương pháp phân tích tương quan

Như chương trước đã giới thiệu, tín hiệu ngẫu nhiên $x(t)$ có thời gian tồn tại hữu hạn phụ thuộc vào τ. Hàm tương quan $B(\tau)$ được tính theo công thức:

$$B_x(\tau) = \int_{-\infty}^{+\infty} x(t)x(t+\tau)dt \tag{2.17}$$

Hàm tương quan phản ánh mối liên hệ giữa tín hiệu và bản thân nó sau khi dịch chuyển một quãng thời gian τ. Thực ra do có sự biến thiên nên ta xét trong quá trình dừng theo nghĩa rộng thì hàm $B_x(\tau)$ được tính như giá trị trung bình của $x(t)$ và $x(t+\tau)$ tức là

$$B_x(\tau) = \overline{x(t)x(t+\tau)} = \underset{T\to+\infty}{Lim}\frac{1}{T}\int_{-\infty}^{+\infty} x(t)x(t+\tau)dt \qquad (2.18)$$

Hàm tương quan có một số tính chất như sau:

1. Hàm tương quan là một hàm chẵn

$$B_x(\tau) = B_x(-\tau).$$

2. Trị số hàm tương quan khi $\tau = 0$ trùng với công suất trung bình của quá trình:

$$B_x(0) = \overline{x^2(t)} = \int_{-\infty}^{+\infty} x^2(t)dt\,.$$

3. Giá trị hàm tương quan khi $\tau = 0$ đạt giá trị cực đại

$$B_x(0) \geq B_x(\tau), \forall\, \tau.$$

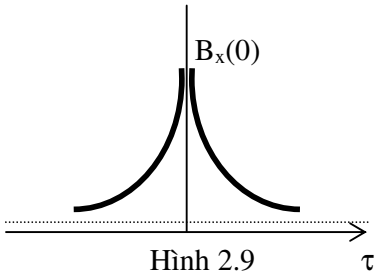
4. Nếu hàm tương quan thỏa mãn điều kiện

$$B_x(\tau) = \begin{cases} \neq 0, & \tau = 0, \\ 0, & \tau \neq 0. \end{cases}$$

thì giữa $x(t)$ và $x(t+\tau)$ không tồn tại tương quan thống kê

5. Khi $\tau \to \infty$ thì giữa $x(t)$ và $x(t+\tau)$ sẽ độc lập với nhau khi đó hàm tương quan sẽ dần tới 0.

Đồ thị mô tả hàm tương quan có dạng như hình vẽ



Hình 2.9

Phương pháp phân tích phổ

Quan sát các quá trình ngẫu nhiên ta chỉ có thể xác lập được phổ chạy

$$S_T(\omega) = \int_0^T S(t)e^{-j\omega t}dt \tag{2.18}$$

Hàm tương quan:

$$B(\tau) = \int_{-\infty}^{T} S(t)e^{-j\omega t}dF(\omega) \tag{2.19}$$

Trong đó $\frac{dF}{d\omega} = \frac{1}{2\pi}G(\omega)$

Người ta gọi G(ω) là phổ năng lượng, khi đó

$$B(\omega) = \frac{1}{2\pi}\int_{-\infty}^{-\infty} G(\omega)e^{j\omega\tau}d\omega \tag{2.20}$$

Trong trường hợp này, G(ω) được xem như là biến đổi Fourier của B(τ)

$$G(\omega) = \int_{-\infty}^{-\infty} B(\tau)e^{-j\omega\tau}d\tau \tag{2.21}$$

Do B(τ) và G(ω) là các hàm chẵn nên chỉ lấy giá trị $\cos(\omega\tau)$ tức là

$$B(\tau) = \frac{1}{\pi}\int_0^{+\infty} G(\omega)\cos\omega\tau d\omega \ ;$$

$$G(\omega) = 2\int_0^{+\infty} B(\tau)\cos\omega\tau d\tau \tag{2.22}$$

Nếu τ = 0 thì:

$$B(0) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} G(\omega)d\omega$$

2.3 Nhiễu trắng

Các hiện tượng xáo động nhiệt trong các phần tử của mạch điện hay dây dẫn, hoặc bức xạ trong khí quyển đều gây ra một loại tín hiệu nhiễu có dải phổ rất rộng gọi là nhiễu trắng. Nhiễu là thành phần không thể bỏ qua khi nghiên cứu về các kênh, nhiễu trắng cũng là một loại tín hiệu ngẫu nhiên. Qua đo đạc

nghiên cứu ta tìm được công thức tính mật độ phân bố xác suất của nhiễu theo quy luật của phân phối chuẩn Gauss

$$W(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{2.23}$$

Trong đó σ được gọi là công suất trung bình của nhiễu. Từ đó ta thấy quy luật phân bố xác suất của nhiễu được xác định bởi hàm phân phối xác suất

$$F(u) = p(x < u) = \frac{1}{2\pi} \int_{-\infty}^{u} e^{-\frac{t^2}{2}} dt = \frac{1}{2}\left(1 + \Phi(u)\right) \tag{2.24}$$

Trong đó $u = \frac{x}{\sigma}$ gọi là trị số tương đối của nhiễu, $\Phi(u) = \frac{2}{\sqrt{2\pi}} \int_{0}^{u} e^{-\frac{t^2}{2}} dt$

Đối với hàm phân phối, ta có các tính chất sau đây

- $\Phi(u)$ là hàm lẻ
- $\Phi(\infty) = 1$, như vậy $\Phi(u)$ có tính hội tụ
- $\Phi(0) = 0$
- $p(u_1 < u < u_2) = \frac{1}{2}\left(\Phi(u_2) - \Phi(u_1)\right)$
- $p(u > u_0) = \frac{1}{2}\left(1 - \Phi(u_0)\right)$

Dùng phương pháp phân tích phổ để khảo sát nhiễu, ta coi nhiễu trắng như một hàm ngẫu nhiên $x(t)$ trong khoảng $(-\infty; +\infty)$. Xét trong một đoạn đủ dài $\left(-\frac{T}{2}, \frac{T}{2}\right)$ có k xung. Người ta đã phân tích và thu được kết quả:

$$G(\omega) = \underset{T \to +\infty}{Lim} \frac{k}{T} 2\left|S(\omega)\right|^2 = 2k_1 \left|S(\omega)\right|^2 \text{ trong đó } k_1 = \underset{T \to +\infty}{Lim} \frac{k}{T}$$

Khi đó ta gọi $G(\omega)$ là phổ năng lượng của nhiễu được xác định theo $S(\omega)$ của từng xung, trong thực tế nhiễu đến một giá trị nào đó sẽ giảm nhỏ khi $\omega \to +\infty$

CHƯƠNG 3. LƯỢNG TIN, ENTROPI NGUỒN RỜI RẠC

3.1 Độ đo thông tin

3.1.1 Khái niệm độ đo

Đối với một đại lượng vật lý bất kỳ, để nghiên cứu về đại lượng đó chúng ta phải trang bị một đơn vị xác định độ lớn của đại lượng đó được gọi là độ đo. Mỗi độ đo phải thỏa mãn 3 tính chất sau:

- Độ đo là một đại lượng không âm.
- Độ đo phải cho phép ta xác định được độ lớn của đại lượng đó. Đại lượng càng lớn, giá trị đo được phải càng cao.
- Độ đo phải tuyến tính: tức là giá trị đo được của đại lượng tổng cộng phải bằng tổng giá trị của các đại lượng riêng phần khi sử dụng độ đo này để đo chúng.

3.1.2 Độ đo thông tin.

Khi nghiên cứu về thông tin, hiển nhiên đây cũng là một đại lượng vật lý, vì vậy chúng ta cũng phải xác định một độ đo cho thông tin. Để xây dựng độ đo cho thông tin chúng ta cần chú ý một số vấn đề sau đây:

Theo bản chất của thông tin thì hiển nhiên thông tin càng có ý nghĩa khi nó càng ít xuất hiện, nên độ đo của nó phải tỷ lệ nghịch với xác suất xuất hiện của tin hay nói cách khác hàm độ đo phải là hàm tỉ lệ nghịch với xác suất xuất hiện của tin tức.

Kí hiệu x là tin với xác suất xuất hiện là $p(x)$. Khi đó hàm độ đo kí hiệu là $I(x) = f\left(\frac{1}{p(x)}\right)$ là hàm tỉ lệ nghịch với xác suất $p(x)$.

Một tin x sẽ là không có ý nghĩa nếu chúng ta đã biết về nó hay xác suất xuất hiện $p(x) = 1$. Trong trường hợp này độ đo phải bằng không tức là $I(x) = 0$.

Xét 2 tin x, y là độc lập thống kê với xác suất xuất hiện tương ứng là $p(x), p(y)$ khi đó tin $z = xy$ là tin khi xuất hiện đồng thời 2 tin x, y cùng một thời điểm. Do đó theo tính chất tuyến tính, chúng ta phải có

$$I(xy) = I(x) + I(y).$$

Như vậy để xây dựng hàm độ đo thông tin, ta thấy hàm $I(x)$ phải là hàm không âm và thỏa mãn đồng thời cả 3 điều kiện đã nêu. Dễ thấy trong tất cả các hàm toán học đã biết thì nếu chọn

$$I(x) = \log_a\left(\frac{1}{p(x)}\right), a > 1$$

thì tất cả các điều kiện đều được thỏa mãn bởi vì

$$I(x) = \log_a\left(\frac{1}{p(x)}\right) \geq 0, a > 1, \forall\, 0 \leq p(x) \leq 1.$$

$I(x) = \log_a\left(\frac{1}{p(x)}\right), a > 1$ là hàm số nghịch biến với xác suất $p(x)$.

$$I(x) = \log_a\left(\frac{1}{p(x)}\right) = 0,\ p(x) \equiv 1.$$

$$I(xy) = \log_a\left(\frac{1}{p(xy)}\right) = \log_a\left(\frac{1}{p(x)p(y)}\right)$$

$$= \log_a\left(\frac{1}{p(x)}\right) + \log_a\left(\frac{1}{p(y)}\right) = I(x) + I(y) \text{ với } x, y \text{ độc lập.}$$

Xuất phát từ những lý do trên, trong lý thuyết thông tin, hàm số

$$I(x) = \log_a\left(\frac{1}{p(x)}\right) = -\log_a(p(x)), a > 1 \qquad (3.1)$$

được chọn làm độ đo thông tin hay lượng đo thông tin của một tin của nguồn.

Trong công thức xác định độ đo thông tin này, cơ số của hàm logarit có thể chọn tùy ý thỏa mãn $(a > 1)$ tuy nhiên người ta thường dùng các đơn vị đo như sau:

- Bit hay đơn vị nhị phân khi cơ số là 2.
- Nat hay đơn vị tự nhiên khi cơ số là e.
- Hartley hay đơn vị thập phân khi cơ số là 10.

3.2 Lượng tin của nguồn rời rạc

3.2.1 Mối liên hệ của lượng tin và lý thuyết xác suất

Khái niệm thông tin là một khái niệm được hình thành từ lâu trong tư duy của con người. Để diễn tả khái niệm này, ta giả thiết rằng trong một tình huống nào đó, có thể xảy ra nhiều sự kiện khác nhau và việc xảy ra một sự kiện nào đó trong tập hợp các sự kiện có thể làm cho ta thu nhận được thông tin.

Một tin đối với người nhận có hai phần

- Độ bất ngờ của tin.
- Ý nghĩa của tin.

Để so sánh các tin với nhau, ta có thể lấy một trong hai hoặc cả hai nội dung trên làm thước đo. Nhưng nội dung hay ý nghĩa của tin mà ta còn gọi là tính hàm ý của tin, không ảnh hưởng đến các vấn đề cơ bản của hệ thống truyền tin như tốc độ hay độ chính xác. Nó chính là ý nghĩa của những tin mà con người muốn trao đổi với nhau thông qua việc truyền tin.

Độ bất ngờ của tin liên quan đến các vấn đề cơ bản của hệ thống truyền tin. Ví dụ: một tin càng bất ngờ, sự xuất hiện của nó càng hiếm, thì rõ ràng thời gian nó chiếm trong một hệ thống truyền tin càng ít.

Như vậy, muốn cho việc truyền tin có hiệu suất cao thì không thể coi các tin như nhau nếu chúng xuất hiện ít nhiều khác nhau.

Để định lượng thông tin trong các hệ thống truyền tin, ta lấy độ bất ngờ của tin để so sánh các tin với nhau. Ta quy ước rằng lượng tin càng lớn nếu độ bất ngờ của tin càng cao. Điều này là hợp lý vì khi ta nhận được một tin đã biết trước thì xem như không nhận được gì, và việc nhận được một tin mà ta ít có hy vọng nhận được thì lại rất quý đối với chúng ta.

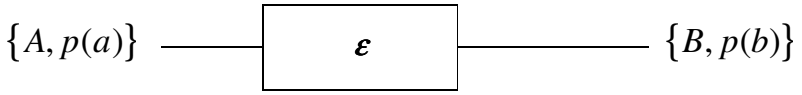
Mỗi tin tức được thể hiện qua mỗi sự kiện. Các sự kiện là các hiện tượng ngẫu nhiên có thể được mô tả bởi các quy luật thống kê.

Về mặt truyền tin ta chỉ quan tâm đến độ bất ngờ của tin hay xác suất xuất hiện các ký hiệu. Để nghiên cứu vấn đề này ta dùng các quy luật thống kê. Phép biến đổi tổng quát trong hệ thống truyền tin là phép biến đổi cấu trúc thống kê của nguồn

Bây giờ chúng ta xem xét mối liên hệ giữa khái niệm tin tức với lý thuyết xác suất. Một nguồn tin rời rạc được xem như một tập hợp các tin $x^{(k)}$ hình thành bởi những dãy ký hiệu hữu hạn x_i là một ký hiệu a_i bất kỳ thuộc nguồn A được gửi đi ở thời điểm t_j. Tin $x^{(k)}$ có dạng: $x^{(k)} = (x_1, x_2, ..., x_n)$ với xác suất xuất hiện $p(x^{(k)})$.

Về mặt toán học nguồn tin X cũng đồng nghĩa với một trường xác suất hữu hạn gồm các điểm $x^{(k)}$. $(k = 1,2,...,M)$ trong không gian n chiều. M là tổng số các điểm được tính bằng $M = m^n$.

Phép biến đổi tổng quát trong một hệ thống truyền tin là phép biến đổi có cấu trúc thống kê của nguồn. Chúng ta có thể lấy bất kỳ một khâu xử lý tin tức nào đó trong hệ thống như rời rạc hóa, mã hóa, điều chế, truyền lan, giải điều chế, giải mã đều có thể xem như một phép biến đổi nguồn. Nói cách khác phép xử lý đó đã biến đổi cấu trúc thống kê của tập tin ở đầu vào khâu hệ thống trở thành một tập tin mới với một cấu trúc thống kê mong muốn ở đầu ra.



Hình 3.1

Trong đó

- $\{A, p(a)\}$ là nguồn ở đầu vào với bộ chữ A và phân bố xác suất các ký hiệu $p(a)$.
- $\{B, p(b)\}$ là nguồn ở đầu ra với bộ chữ B và phân bố xác suất các ký hiệu $p(b)$.

Nếu $\boldsymbol{\varepsilon}$ là quy luật biến đổi thì ta có mối quan hệ $\boldsymbol{\varepsilon} = \{B, p(b)\}$.

Chúng ta có thể mô tả nguồn tin ở đầu vào bằng tập tin $U = \{u^{(i)}\}$ và quy luật phân bố xác suất các tin $p(u^{(i)})$. Trong đó $u^{(i)} = (u_1, u_2, ..., u_n)$, u_k là các tin thuộc A xảy ra ở các thời điểm t_k.

Tương tự nguồn tin ở đầu ra được mô tả bằng tập tin $V = \{v^{(i)}\}$ với quy luật phân bố xác suất $p(v^{(i)})$. Trong đó $v^{(i)} = (v_1, v_2, ..., v_n)$, v_k là một ký hiệu thuộc bộ chữ B xảy ra tuần tự ở thời điểm t_k.

Các tin $u^{(i)}$ hay $v^{(j)}$ được xem như những phần tử của tập U hay V; hoặc những bộ của tập tích Đề các của n tập.

Như vậy $u^{(i)}$ và $v^{(j)}$ lần lượt là phần tử của tập:

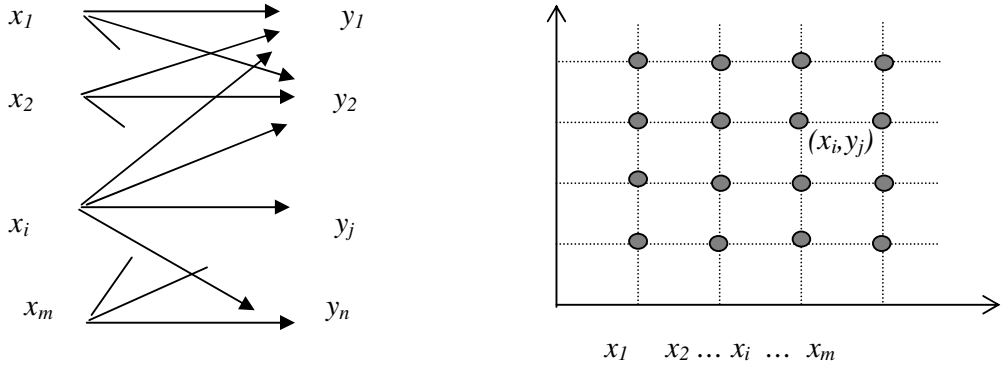
$$U = X \times Y \times Z ... \quad \text{với } X = Y = Z = ... = A$$

$$V = O \times P \times Q ... \quad \text{với } O = P = Q = ... = B$$

Nguồn tin được xem như không gian điểm rời rạc nhiều chiều, mỗi một điểm đại diện cho một tin. Phép biến đổi nguồn chuyển một không gian tin này sang một không gian tin khác. Ví dụ phép rời rạc hóa, chuyển một không gian tin liên tục thành không gian tin rời rạc.

Phép biến đổi trong kênh cũng có thể được xem như những phép biến đổi nguồn khác, tuy nhiên vì có tác động của nhiễu nên sự chuyển đổi giữa các tin thông thường không phải là một – một

Kết quả biến đổi các tin trong kênh có thể được xem như các phần tử của tập tích $X.Y$. Quy luật phân bố xác suất các tin $p(x, y)$ của tập tích $X.Y$ tùy thuộc vào quy luật phân bố xác suất của tập vào $p(x)$ và tính chất thống kê của kênh nghĩa là xác suất chuyển đổi từ tin x thành tin y: $p(y \mid x)$. $p(xy) = p(x)p(y \mid x)$. Nếu có nguồn tin với số ký hiệu bất kỳ $X = \{x_1, x_2, ..., x_m\}$. Đầu ra thu được nguồn $Y = \{y_1, y_2, ..., y_n\}$. Ta xét như sau:



Hình 3.2

Phép biến đổi trong kênh tạo ra một nguồn mới $U = X.Y$, với các tin là các cặp (x_i, y_j), trong đó $x_i \in X$, $y_j \in Y$, theo quy luật phân bố xác suất $p(x_i, y_j)$. Các tin (x_i, y_j) là các điểm rời rạc trên mặt phẳng XY.

Theo lý thuyết xác suất, sự liên hệ giữa các xác suất của các phần tử trong tập X, Y và $U = X.Y$ có thể tính như sau:

$$p(x) = \sum_{y \in Y} p(x, y); \; p(y) = \sum_{x \in X} p(x, y)$$

$$p(x, y) = p(x)p(y/x) = p(y)p(x/y)$$

$$p(x/y) = \frac{p(x)p(y/x)}{\sum_{y \in Y} p(x)p(y/x)}$$

Ví dụ 1: Phép mã hóa nhị phân, cho một nguồn tin $U = \{u_0, u_1, ..., u_7\}$ dùng mã nhị phân để mã hóa nguồn tin, với phép mã hóa như sau:

$$u_0 \rightarrow x_0 y_0 z_0$$

$$u_1 \rightarrow x_1 y_0 z_0$$

$$u_2 \rightarrow x_0 y_1 z_0$$

$$u_3 \rightarrow x_1 y_1 z_0$$

$$u_4 \rightarrow x_0 y_0 z_1$$

$$u_5 \rightarrow x_1 y_0 z_1$$

$$u_6 \rightarrow x_0 y_1 z_1$$

$$u_7 \rightarrow x_1 y_1 z_1$$

trong đó $x_0 = y_0 = z_0 = 0$; $x_1 = y_1 = z_1 = 1$; các mã hiệu thiết lập như trên là các phần tử của một tập tích $X.Y.Z$ và được đại biểu bằng những điểm rời rạc trong một không gian 3 chiều.

Sự liên hệ giữa quy luật phân bố xác suất trong các tập hợp và tập tích đã cho trong lý thuyết xác suất như sau:

$$p(x) = \sum_{y \in Y} p(x, y) = \sum_{y \in Y; z \in Z} p(x, y, z)$$

$$p(y) = \sum_{x \in X} p(x, y) = \sum_{x \in X; z \in Z} p(x, y, z)$$

$$p(z) = \sum_{x \in X} p(x, z) = \sum_{x \in X; y \in Y} p(x, y, z)$$

$$p(x, y) = \sum_{z \in Z} p(x, y, z)$$

$$p(x, z) = \sum_{y \in Y} p(x, y, z)$$

$$p(y, z) = \sum_{x \in X} p(x, y, z)$$

$$p(x, y, z) = p(x)p(yz / x) = p(y)p(xz / y) = p(z)p(xy / z)$$

Áp dụng các biểu thức trên trong việc xác định xác suất của mã hiệu, khi nhận được ở đầu ra của bộ mã hóa lần lượt các ký hiệu của một dãy nào đó. Giả sử ở đầu ra nhận được dãy $x_1 y_0 z_1$. Hãy tính xác suất của tin sau khi nhận được lần lượt các ký hiệu của dãy.

Xác suất của tin u_i sau khi nhận được ký hiệu x_1 được tính theo xác suất có điều kiện $p(y, z / x_1) = \dfrac{p(x_1, y, z)}{p(x_1)}$ trong đó

$$p(x_1) = p(u_1) + p(u_3) + p(u_5) + p(u_7) = \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16} = \frac{1}{2}$$

Xác suất của tin u_i sau khi nhận được ký hiệu x_1, y_0 tính theo xác suất có điều kiện sau: $p(z / x_1, y_0) = \dfrac{p(x_1, y_0, z)}{p(x_1, y_0)}$ trong đó $p(x_1, y_0) = \dfrac{1}{4} + \dfrac{1}{16} = \dfrac{5}{16}$

Xác suất của tin u_i sau khi nhận được ký hiệu x_1, y_0, z_1 chỉ có khả năng xảy ra là u_5: $p(u_5 / x_1, y_0, z_1) = \dfrac{p(x_1, y_0, z_1)}{p(x_1, y_0, z_1)} = 1$, còn lại các tin khác đều có xác suất bằng 0. Kết quả tính toán được cho trong bảng 3.1

u_i	$p(u_i)$	XYZ	Xác suất của tin sau khi nhận được ký hiệu		
			x_1	y_0	z_1
u_0	1/4	$x_0y_0z_0$	0	0	0
u_1	1/4	$x_1y_0z_0$	1/2	4/5	0
u_2	1/8	$x_0y_1z_0$	0	0	0
u_3	1/8	$x_1y_1z_0$	1/4	0	0
u_4	1/16	$x_0y_0z_1$	0	0	0
u_5	1/16	$x_1y_0z_1$	1/8	1/5	1
u_6	1/16	$x_0y_1z_1$	0	0	0
u_7	1/16	$x_1y_1z_1$	1/8	0	0

Bảng 3.1

3.2.2 Lượng tin riêng, lượng tin tương hỗ, lượng tin có điều kiện

Như trong phần trước ta đã đề cập về độ đo thông tin, hàm logarit đã được chọn để đánh giá, định lượng các lượng tin. Đối với mỗi tin x_i của nguồn X đều có lượng tin riêng như đã biết:

$$I(x_i) = \log\left(\frac{1}{p(x_i)}\right)$$

Nếu nguồn X thông qua một phép biến đổi trở thành nguồn Y ví dụ thông qua sự truyền lan trong kênh thì phép biến đổi đó có thể không phải là 1-1.

Ở đầu vào của kênh là các tin $x_i \in X$, các tin trong quá trình truyền lan trong kênh bị nhiễu phá hoại, làm cho sự chuyển đổi từ nguồn X sang nguồn Y không phải là 1-1 . Một tin $x_i \in X$ có thể chuyển thành một tin $y_j \in Y$ ở đầu ra của kênh với những xác suất chuyển đổi khác nhau tùy thuộc theo tính chất nhiễu trong kênh.

Bài toán truyền tin trong trường hợp này đặt ra là: Cho biết cấu trúc thống kê của nguồn X, tính chất tạp nhiễu của kênh biểu thị dưới dạng các xác suất chuyển đổi của tin, khi nhận được một tin $y_j \in Y$, hãy xác định tin tương ứng của nguồn X.

Đây là bài toán thống kê, lời giải khẳng định là không có được. Lời giải tìm được sẽ có dạng: với tin $y_j \in Y$ nhận được, tin nào của nguồn X có nhiều khả năng đã được phát đi nhất.

Muốn giải quyết vấn đề này ta lần lượt qua hai bước

Bước 1: Tính các lượng tin về một tin bất kỳ $x_i \in X$ chứa trong tin $y_j \in Y$ nhận được, lượng tin đó gọi là lượng tin tương hỗ giữa x_i và y_j.

Muốn xác định lượng tin tương hỗ ta phải tìm lượng tin ban đầu có trong x_i, sau khi thực hiện quá trình truyền tin ta cần xác định tìm lượng tin còn lại trong x_i, hiệu hai lượng tin này cho ta thấy lượng tin đã truyền từ x_i sang y_j.

Lượng tin ban đầu là lượng tin riêng được xác định bằng xác suất tiên nghiệm của tin: $I(x_i) = \log\left(\frac{1}{p(x_i)}\right)$.

Lượng tin bị nhiễu phá hủy một phần được xác định bằng xác suất hậu nghiệm $I(x_i \mid y_j) = \log\frac{1}{p(x_i \mid y_j)}$, lượng tin này còn gọi là lượng tin có điều kiện, trong quá trình truyền tin, lượng tin đó chính là lượng tin đã bị tạp nhiễu phá hủy không đến đầu thu được.

Như vậy lượng tin tương hỗ được tính theo công thức sau:

$$I(x_i, y_j) = I(x_i) - I(x_i \mid y_j) = \log\frac{p(x_i \mid y_j)}{p(x_i)}$$

$$= \log\left[p(x_i \mid y_j) / \sum_j p(y_j)\, p(x_i \mid y_j) \right]$$

Bước 2: Đem so sánh các lượng tin tương hỗ với nhau, và lượng tin nào cực đại sẽ cho biết tin x_i có khả năng nhiều nhất chuyển thành y_j trong quá trình truyền tin.

Trong trường hợp phức tạp

Mở rộng khái niệm lượng tin tương hỗ trong trường hợp mã hóa hay phép biến đổi phức tạp hơn. Lúc đó lượng tin tương hỗ cũng được xác định theo các công thức xác suất tiên nghiệm và xác suất hậu nghiệm của tin đang xét. Ví dụ một phép biến đổi kép $X \to Y \to Z$.

Lượng tin ban đầu của x_i được xác định theo xác suất ban đầu hay xác suất tiên nghiệm. Sau khi nhận được tin y_j, xác suất của tin x_i trở thành xác suất có điều kiện $p(x_i \mid y_j)$, và cuối cùng khi đã nhận được y_j và z_k thì xác suất của x_i là $p(x_i \mid y_j z_k)$.

Nếu ta xem $p(x_i)$ là xác suất tiên nghiệm và $p(x_i \mid y_j)$ là xác suất hậu nghiệm thì ta lại trở lại trường hợp biến đổi đơn giản đã nói ở trên và có lượng tin tương hỗ giữa x_i và y_j: $I(x_i, y_j)$.

Nếu ta xem $p(x_i \mid y_j)$ là xác suất tiên nghiệm và $p(x_i \mid y_j z_k)$ là xác suất hậu nghiệm, ta sẽ xác định được lượng tin tương hỗ giữa x_i và z_k với điều kiện đã biết y_j: $I(x_i, z_k \mid y_j) = \log \dfrac{p(x_i \mid y_j z_k)}{p(x_i \mid y_j)}$

Nếu ta xem $p(x_i)$ là xác suất tiên nghiệm và $p(x_i \mid y_j z_k)$ là xác suất hậu nghiệm thì lượng tin tương hỗ giữa x_i và cặp (y_j, z_k) là:

$$I(x_i, y_j z_k) = \log \frac{p(x_i \mid y_j z_k)}{p(x_i)}$$

Một cách trực giác có thể nhận thấy lượng tin về x_i chứa trong cặp (y_j, z_k) phải bằng lượng tin về x_i chứa trong y_j cộng với lượng tin tương hỗ về x_i chứa trong z_k khi đã biết y_j. Điều này có thể được xác minh một cách dễ dàng như sau:

$$I(x_i, y_j z_k) = \log \frac{p(x_i \mid y_j z_k)}{p(x_i)} \frac{p(x_i \mid y_j)}{p(x_i \mid y_j)} = I(x_i, y_j) + I(x_i, z_k \mid y_j)$$

Nếu thay thứ tự các tập X, Y, Z thì sẽ có các biểu thức mới nhưng ý nghĩa hoàn toàn không thay đổi.

Ví dụ: Tính lượng tin tương hỗ giữa tin u_5 và các ký hiệu lần lượt nhận được, trong ví dụ mã hóa nhị phân đã nêu trong ví dụ trước

$$I(u_5, x_1) = \log\frac{1/8}{1/16} = \log 2$$

$$I(u_5, y_0 \mid x_1) = \log\frac{1/5}{1/8} = \log\frac{8}{5}$$

$$I(u_5, z_1 \mid x_1 y_0) = \log\frac{1}{1/5} = \log 5$$

Tổng lượng tin về u_5 được biết lần lượt khi nhận x_1, y_0, z_1: log16

3.2.3 Tính chất của lượng tin

Tính chất 1: Lượng tin riêng là một đại lượng không âm (vì $p(x_i) \le 1$ nên $-\log p(x_i) \ge 0$). Nhưng lượng tin tương hỗ có thể dương, có thể âm do phụ thuộc lượng tin có điều kiện.

Tính chất 2: Lượng tin riêng bao giờ cũng lớn hơn lượng tin về nó chứa trong bất kỳ ký hiệu nào có liên hệ thống kê với nó. Do vậy khi x_i và y độc lập thống kê thì lượng tin tương hỗ bằng 0. Lượng tin tương hỗ cực đại khi $p(y_j \mid x_i) = 1$ và bằng lượng tin riêng.

$$I(x_i, y_j) = \log\frac{p(x_i \mid y_j)}{p(x_i)} = \log\frac{p(y_j \mid x_i)}{p(y_j)} \le I(x_i) = \log\frac{1}{p(x_i)}$$

Điều nói trên cho thấy lượng tin tương hỗ mô tả sự ràng buộc giữa x_i và y_j, nếu sự ràng buộc ấy càng chặt chẽ thì lượng tin về x_i chứa trong y_j càng lớn, hay lượng tin về y_j chứa trong x_i cũng tăng lên. Từ đó cũng có thể giải thích ý nghĩa của lượng tin như là lượng tin tương hỗ cực đại giữa x_i và y_j.

Tính chất 3: Lượng tin của một cặp ($x_i y_j$) bằng tổng lượng tin riêng của từng tin trừ đi lượng tin tương hỗ giữa chúng.

$$I(x_i y_j) = I(x_i) + I(y_j) - I(x_i, y_j)$$

Khi x_i và y_j độc lập thống kê $I(x_i, y_j) = 0$

Đối với trường hợp nguồn phức tạp U=XYZ:

$$I(x_i \mid z_k) = -\log p(x_i \mid z_k) \geq I(x_i, y_j \mid z_k).$$

Giải thích lượng tin riêng có điều kiện $I(x_i \mid z_k)$, $I(y_j \mid z_k)$ cũng tương tự như giải thích lượng tin riêng. Lượng tin riêng có điều kiện chính là lượng tin tương hỗ với cùng một điều kiện đã xác định một cách đơn trị giữa các tin với nhau. Lượng tin tương hỗ có thể phân thành tổng của những lượng tin tương hỗ khác:

$$I(x_i, y_j z_k) = I(x_i, y_j) + I(x_i, z_k \mid y_j)$$

3.2.4 Lượng tin trung bình

Lượng tin riêng chỉ có ý nghĩa đối với một tin nào đó, nhưng không phản ánh được giá trị tin tức của nguồn. Nói một cách khác $I(x_i)$ chỉ đánh giá được về mặt tin tức của một tin khi nó đứng riêng rẽ, nhưng không thể dùng để đánh giá về mặt tin tức của tập hợp trong đó x_i tham gia. Trong thực tế điều ta quan tâm là giá trị tin tức của một tập hợp chứ không phải giá trị tin tức một phần tử nào đó trong tập hợp. Để đánh giá hoàn chỉnh giá trị tin tức của một tin x_i trong cả bảng tin ta dùng khái niệm lượng tin trung bình.

Định nghĩa*:* Lượng tin trung bình là lượng tin tức trung bình chứa trong một ký hiệu bất kỳ của nguồn đó cho.

$$I(X) = -\sum_{x \in X} p(x) \log p(x) \tag{3.6}$$

Ví dụ: Một nguồn tin có hai ký hiệu là x_0, x_1 với xác suất $p(x_0) = 99\%$ và $p(x_1) = 1\%$. Như vậy khi nhận một tin ta biết gần như chắc chắn là đó là x_0, tin này không còn bất ngờ nên giá trị tin tức rất nhỏ. Thế nhưng xét lượng tin riêng của x_1:

$$I(x_1) = -\log 0,01 \approx 6,64 \text{ (bit/kí hiệu)}$$

Đó là giá trị rất lớn, điều đó không phản ánh đúng giá trị của tin như đã xét ở trên. Nếu xét lượng tin trung bình:

$$I(X) = -p(x_0) \log p(x_0) - p(x_1) \log p(x_1) = 0,066 + 0,014 = 0,08 \text{ (Bit/kí hiệu)}$$

Như vậy lượng tin trung bình rất nhỏ phản ánh đúng thực tế giá trị của nguồn tin.

Ta cũng có khái niệm lượng tin tương hỗ trung bình:

$$I(X,Y) = \sum_{x\in X;\, y\in Y} p(x,y)\log\frac{p(x\mid y)}{p(x)} \tag{3.7}$$

Lượng tin có điều kiện trung bình:

$$I(X/Y) = \sum_{x\in X;\, y\in Y} p(x,y)\log p(x\mid y) \tag{3.8}$$

Ta có quan hệ giữa các lượng tin trung bình:

$$I(X,Y) = I(X) - I(X\mid Y)$$
$$I(X,Y) = I(Y,X) \geq 0$$

Đối với trường hợp nguồn phức tạp $U = XYZ$

$$I(X,YZ) = \sum_{x\in X;\, y\in Y;\, z\in Z} p(x,y,z)\log\frac{p(x\mid yz)}{p(x)}$$
$$I(X,Y/Z) = \sum_{x\in X;\, y\in Y;\, z\in Z} p(x,y,z)\log\frac{p(x\mid yz)}{p(x\mid y)} \tag{3.9}$$

3.3 Entropi của nguồn rời rạc

3.3.1 Khái niệm entropi

Khi nhận được một tin ta sẽ nhận được một lượng tin trung bình, đồng thời độ bất ngờ về tin đó cũng đã được giải thoát, cho nên độ bất ngờ và lượng tin về ý nghĩa vật lý trái ngược nhau, nhưng về số đo thì giống nhau và được xác định theo công thức sau:

$$H(x) = \log\frac{1}{p(x)}$$

Độ bất ngờ trung bình của một tin thuộc nguồn X (entropi của nguồn) được xác định theo công thức sau:

$$H(X) = -\sum_{x\in X} p(x)\log p(x) \tag{3.11}$$

3.3.2 Tính chất của entropi

Tính chất 1: Entropi là một đại lượng không âm: H(X) ≥ 0

Tính chất 2: H(X) = 0 khi nguồn có một ký hiệu bất kỳ có xác suất xuất hiện bằng 1 và xác suất xuất hiện tất cả các ký hiệu còn lại bằng không. Nghĩa là nguồn có một tin luôn được xác định, như vậy giá trị thông tin của nguồn bằng không.

Tính chất 3: Entropi cực đại khi xác suất xuất hiện của các ký hiệu bằng nhau

Chứng minh:

Các giá trị $p(x)$ làm cực đại hàm $H(X) = -\sum_{x\in X} p(x)\log p(x)$

với điều kiện $\sum_{x\in X} p(x) = 1$ cũng chính là các giá trị $p(x)$ làm cực đại hàm

$\Phi = -\sum_{X} p(x)\log p(x) + \lambda\left(\sum_{X} p(x) - 1\right)$ λ được gọi là nhân tử Lagrange. Các giá trị p(x) làm cực đại hàm Φ thỏa mãn điều kiện:

$\frac{\partial\Phi}{\partial p(x)} = 0$ với mọi giá trị p(x) hay $\frac{\partial\Phi}{\partial p(x)} = -\log p(x) - \log e + \lambda = 0$ với mọi giá trị $p(x)$. Tức là các giá trị $p(x)$ bằng nhau với tất cả các tin của nguồn, và khi đó giá trị cực đại của $H(X)$ sẽ là $\log_2 m$ nếu lấy đơn vị là bit và nguồn có m tin.

Nếu nguồn có m ký hiệu đẳng xác suất thì xác suất xuất hiện một ký hiệu là 1/m khi đó: $H(X) = \log m$

3.3.3 Entropi đồng thời và Entropi có điều kiện

Entropi đồng thời

Entropi đồng thời là độ bất ngờ trung bình của một cặp(x, y) bất kỳ trong tập tích XY. Theo định nghĩa về entropi có:

$$H(X,Y) = -\sum_{x\in X;\, y\in Y} p(x,y)\log p(x,y) \qquad (3.12)$$

Entropi có điều kiện

Khi cần đánh giá sự ràng buộc thống kê giữa các cặp (x, y) ta dùng khái niệm entropi có điều kiện $H(X \mid Y)$ hoặc $H(Y \mid X)$. Đó là độ bất định trung bình của một ký hiệu bất kỳ $x \in X$ khi đó biết bất kỳ một ký hiệu

$y \in Y$. Xuất phát từ các xác suất có điều kiện $p(x \mid y)$ và $p(y \mid x)$ cũng như theo định nghĩa về entropi ta có biểu thức định nghĩa sau:

$$H(X \mid Y) = -\sum_{x \in X;\, y \in Y} p(x, y) \log p(x \mid y)$$
$$H(Y \mid X) = -\sum_{x \in X;\, y \in Y} p(x, y) \log p(y \mid x) \tag{3.13}$$

So sánh với các biểu thức định nghĩa cho các entropi, ta có quan hệ sau:

$$H(X,Y) = H(X) + H(Y \mid X)$$
$$H(X,Y) = H(Y) + H(X \mid Y) \tag{3.14}$$

Trường hợp nguồn phức tạp

Đối với mã hóa hay truyền tin phức tạp hơn ta mở rộng khái niệm entropi cho những tập tích mà số các tập hợp thành nhiều hơn hai, chẳng hạn trường hợp của tập tích XYZ ta có định nghĩa về entropi đồng thời và có điều kiện mở rộng như sau:

$$H(XYZ) = -\sum_{x \in X;\, y \in Y;\, z \in Z} p(x, y, z) \log p(x, y, z)$$
$$H(X \mid YZ) = -\sum_{x \in X;\, y \in Y;\, z \in Z} p(x, y, z) \log p(x \mid y.z) \tag{3.15}$$

3.3.4 Entropi nguồn Markov

Nguồn Markov giữ vai trò quan trọng trong lĩnh vực truyền thông. Nó được đặc trưng bởi quan hệ $p(x_{i_n} \mid x_{j_{n-1}}, x_{k_{n-2}}, ...) = p(x_{i_n} \mid x_{j_{n-1}})$ trong đó x_{i_n} là ký hiệu x_i của nguồn X xuất hiện ở thời điểm *n*. Điều này có nghĩa là xác suất tạo ra một ký hiệu nào đó tại thời điểm n chỉ phụ thuộc vào ký hiệu đã tạo ra ở thời điểm thứ *n-1* và không phụ thuộc vào các ký hiệu đã tạo ra ở các thời điểm *n-2, n-3,…*

Tại thời điểm *n,* nguồn có thể ở trạng thái j với xác suất $p(x_{j_n} / x_{i_{n-1}})$ nào đó khi ở thời điểm *n-1* nguồn đã ở trạng thái *i*.

Xác suất $p(x_{j_n} / x_{i_{n-1}}) = p_{ij}$ gọi là xác suất chuyển đổi từ trạng thái *i* sang trạng thái *j*, trong đó $\sum_{j=1}^{m} p_{ij} = 1$ (*m* là số tin thuộc nguồn).

Xác suất để nguồn ở trạng thái *j* tại thời điểm *n* là:

$$p(x_{j_n}) = \sum_{i=1}^{m} p(x_{i_{n-1}}) p_{ij} \qquad j = 1,2,...,m$$

Biểu diễn dưới dạng ma trận ta có :

$$\mathrm{P}_n = \begin{bmatrix} p(x_{1_n}) \\ p(x_{2_n}) \\ ..... \\ p(x_{m_n}) \end{bmatrix} \quad \mathrm{P}_{n-1} = \begin{bmatrix} p(x_{1_{n-1}}) \\ p(x_{2_{n-1}}) \\ ..... \\ p(x_{m_{n-1}}) \end{bmatrix} \quad \mathrm{T} = \begin{bmatrix} p_{11} & p_{12} & .... & p_{1m} \\ p_{21} & p_{22} & ..... & p_{2m} \\ ... & ..... & ..... & .... \\ p_{m1} & p_{m2} & ..... & p_{mn} \end{bmatrix}$$

Mối quan hệ trên có thể viết : $\mathrm{P}_n = \mathrm{T}^T \mathrm{P}_{n-1}$

Nếu nguồn đang ở trạng thái *i* thì sẽ có một độ bất định về trạng thái của nguồn ở thời điểm sau, đó là trạng thái *j*, trạng thái này là một trong các trạng thái có thể của nguồn. Giá trị trung bình của độ bất định này được xác định bởi entropi $H_i = -\sum_{j=1}^{m} p_{ij} \log p_{ij}$

Nếu tính tới tất cả các trạng thái của nguồn, entropi của nguồn là giá trị trung bình của entropi nguồn *X* ở mỗi trạng thái : $H = \sum p(x_i) H_i$

3.4 Mối quan hệ giữa lượng tin tương hỗ trung bình và Entropi

$$I(X,Y) = \sum_{x \in X; y \in Y} p(x,y) \log \frac{p(x \mid y)}{p(x)} = \sum_{x \in X; y \in Y} p(x,y)(\log p(x \mid y) - \log p(x))$$

$$= \sum_{x \in X, y \in Y} p(x,y) \log p(x \mid y) - \sum_{x \in X, y \in Y} p(x,y) \log p(x)$$

$$= H(X) - H(X \mid Y)$$

Vậy :

$$I(X,Y) = H(X) - H(X \mid Y)$$

$$I(X,Y) = H(Y) - H(Y \mid X)$$

Suy ra $I(X,Y) = H(X) + H(Y) - H(X,Y)$

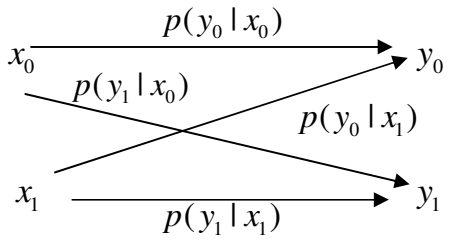
Nếu X,Y độc lập thống kê thì : $I(X,Y) = 0$

Và ta cũng chứng minh được $H(X) \geq H(X \mid Y)$

$$H(Y) \geq H(Y \mid X)$$

Ví dụ :

Cho sơ đồ truyền tin :



Hình 3.3

Biết : $p(x_0) = \frac{1}{4};\ p(x_1) = \frac{3}{4}$

$$p(y_0 \mid x_0) = p(y_1 \mid x_1) = 2/3;\ \ p(y_1 \mid x_0) = p(y_9 \mid x_1) = 1/3$$

+ Tính Entropi đầu vào của kênh :

$$H(X) = -(p(x_0)\log p(x_0) + p(x_1)\log p(x_1)) = 0,81$$

+ Tính Entropi đầu ra :

$$p(y_0) = p(x_0)p(y_0 \mid x_0) + p(x_1)p(y_0 \mid x_1) = 0,417$$

$$p(y_1) = p(x_0)p(y_1 \mid x_0) + p(x_1)p(y_1 \mid x_1) = 0,583$$

$$H(Y) = -(p(y_0)\log p(y_0) + p(y_1)\log p(y_1)) = 0,98$$

+ Tính H(X,Y)

Áp dụng công thức : $H(X,Y) = -\sum_{x \in X;\, y \in Y} p(x,y)\log p(x,y)$

$$p(x_0, y_0) = p(x_0)p(y_0 \mid x_0) = 0,17$$

$$p(x_0, y_1) = p(x_0)p(y_1 \mid x_0) = 0,08$$

$$p(x_1, y_0) = p(x_1)p(y_0 \mid x_1) = 0,25$$

$$p(x_1, y_1) = p(x_1)p(y_1 \mid x_1) = 0,5$$

$$H(X,Y) = 1,73$$

+ Tính $H(X \mid Y)$

$$H(X \mid Y) = H(X,Y) - H(Y) = 1,73 - 0,98 = 0,75$$

3.5 Tốc độ lập tin nguồn rời rạc và thông lượng kênh rời rạc

3.5.1 Tốc độ lập tin

• ***Khái niệm***

Ngoài thông số cơ bản của nguồn là Entropi ta thấy sự hình thành thông tin nhanh hay chậm để đưa vào kênh lại tuỳ thuộc vào bản chất vật lý của nguồn như quán tính, độ phân biệt …

Cho nên số ký hiệu lập được trong một đơn vị thời gian rất khác nhau. Ví dụ : con người vì kết cấu của cơ quan phát âm hạn chế nên một giây chỉ phát âm được từ 5-7 âm tiết trong lời nói thông thường, trong khi máy điện báo có thể tạo ra từ 50-70 ký hiệu trong một giây.

Như vậy thông số thứ hai của nguồn là tốc độ thiết lập tin R (Lượng thông tin nguồn lập được trong một đơn vị thời gian), Tốc độ thiết lập tin tại đầu vào kênh bằng tích của entropi $H(X)$ với số ký hiệu n_0 lập được trong một đơn vị thời gian, trong trường hợp dùng loga cơ số hai thì đơn vị của R là bit/sec.

$$R = n_0 H(X) \tag{3.16}$$

Như vậy muốn nâng cao R thì hoặc là tăng n_0 hoặc là tăng $H(X)$. Tăng n_0 phụ thuộc vào thiết bị phần cứng. Tăng $H(X)$ ta có thể thay đổi cấu trúc thống kê của nguồn như vậy sẽ đơn giản không tốn kém.

Ta đã biết nếu xác suất xuất hiện các ký hiệu bằng nhau thì $H(X)$ cực đại, do vậy ta dùng phép mã hoá để thực hiện việc này mã hoá nguồn tin ban đầu thành nguồn tin mã hoá sao cho xác suất các ký hiệu tương đương nhau.

Ví dụ :

Cho nguồn tin $X = \{x_1, x_2, x_3, x_4\}$ với xác suất tương ứng là :

$$p(x_1) = 1/2; p(x_2) = 1/4; p(x_3) = 1/8; p(x_4) = 1/8$$

$$H(X) = 7/8$$

Nếu có một tin gồm các ký hiệu : $x_1x_1x_4x_2x_1x_2x_1x_3$. Để có $H(X)$ cực đại phải có xác suất các ký hiệu bằng nhau bằng 1/8

Muốn vậy ta mã hoá nguồn X trên thành nguồn Ynhư sau :

$$x_1 = y_0$$
$$x_2 = y_1y_0$$
$$x_3 = y_1y_1y_0$$
$$x_4 = y_1y_1y_1$$

Ta được dãy các ký hiệu của tin : $y_0y_0y_1y_1y_1y_1y_0y_0y_1y_0y_0y_1y_1y_0$

Nguồn này có $H(Y) = 1$

Mã hoá nguồn này thành nguồn Z với các ký hiệu sau :

$$z_1 = y_0y_0$$
$$z_2 = y_0y_1$$
$$z_3 = y_1y_0$$
$$z_4 = y_1y_1$$

Ta được dãy các ký hiệu của tin : $z_1z_4z_4z_1z_3z_2z_3$

Xác suất các ký hiệu của nguồn Z bằng nhau và bằng 1/4 nên :

$$H(Z) = \log 4 = 2$$

Vậy bằng phép mã hoá nguồn X thành nguồn Z ta có thể nâng Entropi của nguồn X là $H(X)$ = 7/4 thành Entropi $H(Z) = 2$ mà vẫn đảm bảo lượng tin trong các bản tin được bảo toàn và có cùng giá trị là 14 (bit)

- **Độ dư của nguồn**

Để chỉ ra sự chênh lệch giữa entropi của nguồn và giá trị cực đại có thể có của nó ta dùng độ dư của nguồn:

$$R = H(X)_{\max} - H(X) \tag{3.17}$$

Ngoài ra cũng có thể dùng độ dư tương đối của nguồn để đánh giá:

$$r = \frac{H(X)_{\max} - H(X)}{H(X)_{\max}} = 1 - \frac{H(X)}{H(X)_{\max}} \tag{3.18}$$

3.5.2 Thông lượng kênh

Thông lượng kênh C là lượng tin cực đại kênh cho đi qua trong một đơn vị thời gian mà không gây ra sai nhầm.
Vậy:

$$C = R_{\max}$$

Đơn vị của thông lượng kênh là bit/giây. Như vậy $R \le C$

Nhiệm vụ của mã hoá thống kê là bằng cách mã hoá để thay đổi Entropi của nguồn để thay đổi tốc độ lập tin R sao cho xấp xỉ với C, gọi là phối hợp giữa nguồn với kênh về phương diện tốc độ truyền tin. Khi truyền tin trong kênh có nhiễu thì nhiệm vụ của mã hóa là lợi dụng điều kiện $R < C$ để xây dựng mã chống nhiễu đồng thời tăng tốc độ lập tin.

Thông lượng kênh rời rạc không có nhiễu

Khi kênh rời rạc không có nhiễu toàn bộ tin tức được thiết lập đều có thể truyền qua kênh mà không bị sai. Vậy ở đầu thu ta nhận được lượng tin bằng với đầu vào hay ta có:

$$C = R_{\max} = n_0 H(X)_{\max}$$

$R_{\max}$ là tốc độ lập tin đầu vào, lượng tin này nhận được nguyên vẹn ở đầu ra. Nếu kênh có $R < C$ thì ta có thể mã hoá để tăng R sao cho:

$C - R < \varepsilon$ với ε nhỏ tuỳ ý

Ta không thể mã hoá cho $R > C$ được, đó là giới hạn của việc mã hoá. Trong trường hợp mã hoá sao cho $R = C$ được gọi là phương pháp mã hoá tối ưu.
Sau khi mã hoá ta có $H(X)_{\max}$. Giữa $H(X)_{\max}$ và $H(X)$ ban đầu ta có độ chênh lệch gọi là độ dư tương đối của nguồn.

$$r = \frac{H(X)_{\max} - H(X)}{H(X)_{\max}} = 1 - \frac{H(X)}{H(X)_{\max}} \qquad (3.19)$$

Vậy phép mã hoá tối ưu cũng có thể coi là phương pháp làm giảm độ dư của nguồn ban đầu.

Thông lượng kênh rời rạc có nhiễu

Thông thường tốc độ lập tin bé hơn nhiều so với thông lượng kênh, nhiệm vụ của mã hóa thống kê là thay đổi tốc độ lập tin của nguồn bằng cách thay đổi entropi, để tốc độ lập tin tiệm cận với thông lượng, gọi là phối hợp với nguồn và kênh về phương diện tốc độ truyền tin.

Trong trường hợp tin nhận được sau không phụ thuộc những tin nhận được trước, nói cách khác chúng độc lập thống kê với nhau thì độ chính xác của tin truyền đi trong kênh chỉ còn bị ảnh hưởng của nhiễu là giảm đi, khi đó tốc độ lập tin tại đầu ra của kênh được định nghĩa như sau:

$$R = n_0 I(X,Y) = n_0 (H(X) - H(X \mid Y)) \tag{3.20}$$

$n_0 H(X \mid Y)$ về mặt độ lớn là lượng tin bị nhiễu phá hủy trong một đơn vị thời gian, vậy muốn nâng cao tốc độ lập tin thì nhất thiết phải thay đổi thông số của nguồn. Lúc đó lượng tin tối đa mà kênh cho đi qua không xảy ra sai nhầm sẽ là tốc độ lập tin cực đại trong kênh có nhiễu:

$$C = n_0 (H(X) - H(X \mid Y))_{\max} \tag{3.21}$$

Độ dư tương đối còn có thể được xác định theo công thức sau: $r_c = 1 - \frac{R}{C}$; Hiệu quả sử dụng kênh: $\lambda_c = 1 - r_c$

CHƯƠNG 4. LÝ THUYẾT MÃ

4.1 Khái niệm mã và điều kiện thiết lập mã

Trong các hệ thống truyền tin rời rạc hoặc truyền các tín hiệu liên tục nhưng đã được rời rạc hóa, bản tin thường phải thông qua một số phép biến đổi hay còn gọi là mã hóa ở phía nguồn phát, phía thu tin cần phải thông qua những phép biến đổi ngược lại là giải mã.

Sự mã hóa thông tin cho phép ta ký hiệu hóa thông tin hay sử dụng các ký hiệu quy ước để biểu diễn bản tin ở dạng phù hợp cho nơi sử dụng. Chính nhờ mã hóa, ta có thể hiển thị được thông tin, thông tin có bản chất là các khái niệm. Đối với một hệ thống truyền tin, việc mã hóa cho phép tăng tính hữu hiệu và độ tin cậy của hệ thống truyền tin, nghĩa là tăng tốc độ truyền tin và tăng khả năng chống nhiễu của hệ thống.

Khi tốc độ lập tin R của nguồn còn cách xa thông lượng C của kênh, nhiệm vụ của mã hóa là biến đổi tính thống kê của nguồn làm cho tốc độ lập tin tiếp cận với khả năng truyền của kênh. Trong trường hợp truyền tin trong kênh có nhiễu, điều cần quan tâm nhiều là độ chính xác của sự truyền tin, hay các tin truyền đi ít bị sai nhầm. Đây chính là nhiệm vụ thứ hai của mã hóa.

Trong chương này, trước tiên ta đề cập tới các khái niệm và định nghĩa về mã: thế nào là mã hiệu? các thông số cơ bản của mã hiệu là gì? các điều kiện và yêu cầu đối với mã hiệu là gì?

4.1.1 Mã hiệu và các thông số cơ bản

Định nghĩa: Mã hiệu là một nguồn tin với một sơ đồ thống kê được xây dựng nhằm thoả mãn một số yêu cầu do hệ thống truyền tin đặt ra như tăng tốc độ lập tin, tăng độ chính xác cho các tin …

Như vậy mã hiệu chính là một tập hữu hạn các ký hiệu riêng hay bảng chữ riêng có phân bố xác suất thỏa mãn một số yêu cầu quy định.

- Việc mã hoá là phép biến đổi 1 – 1 giữa các tin của nguồn được mã hoá với các từ mã do các dấu mã tạo thành.

Cho nguồn S (A,P) với A- tập kí hiệu nguồn, P-xác suất tương ứng. Khi đó phép mã hóa là song ánh f: A -> M, M là tập các từ mã tương ứng.

- Số các ký hiệu khác nhau trong bảng chữ của mã gọi là cơ số mã (m) mỗi ký hiệu có một số trị nào đó tuỳ theo cấu trúc của bộ mã (ví dụ mã nhị phân $m = 2$ sử dụng hai ký hiệu là 0 và 1, đó là loại mã được dùng rộng rãi nhất.

- Số các ký hiệu trong một từ mã gọi là độ dài từ mã n. Nếu các từ mã trong bộ mã có độ dài bằng nhau gọi là mã đồng đều, độ dài từ mã không bằng nhau gọi là mã không đều. Mã không đều ta có khái niệm độ dài trung bình của từ mã tính như sau:

$$\bar{n} = \sum_{i=1}^{N} p(x_i) n_i \qquad (4.1)$$

Trong đó: $p(x_i)$: Xác suất xuất hiện của tin xi được mã hoá thành từ mã thứ i

n_i: Độ dài từ mã ứng với tin xi

N: Tổng số từ mã của bộ mã tương ứng với tổng số các tin x_i

Số từ mã N chính là tổng số các từ mã có trong một bộ mã sau khi mã hoá một nguồn nào đó. Với bộ mã đều theo lý thuyết ta có

$$N = m^n$$

Ví dụ: Một bộ mã nhị phân mỗi từ mã có độ dài 5 bit ta có

$$N = 2^5 = 32 \text{ từ}$$

Nếu ta dùng hết các từ mã hay $N = m^n$ gọi là mã đầy, nếu ta dùng số từ mã $N < m^n$ gọi là mã vơi. Như vậy trong thực tế sử dụng ta thấy có thể có hai loại mã. Với mã đều các từ mã hay bị sai nhầm giữa từ này với từ khác nên dùng bộ mã này ta phải nghiên cứu cách phát hiện sai và sửa sai. Với mã không đều ta phải chọn độ dài các từ mã sao cho độ dài trung bình của từ mã là ngắn nhất gọi là mã thống kê tối ưu.

- Giá trị riêng hay còn gọi là trị của mỗi ký hiệu mã. Mỗi mã hiệu có m ký hiệu mã khác nhau, nếu cơ số của nó là m. Mỗi ký hiệu mã là một dấu hiệu riêng, và chúng được gán một giá trị xác định theo một độ đo xác định, các giá trị này được gọi là các giá trị riêng của các ký hiệu mã và ký hiệu là a, trong trường hợp số, mỗi ký hiệu mã được gán một giá trị riêng nằm trong khoảng từ 0 tới m-1.

- Chỉ số vị trí của ký hiệu mã trong từ mã: ta gọi một vị trí mã là một chỗ trong từ mã để đặt một ký hiệu mã vào đó. Một từ mã có n ký hiệu mã sẽ có n

vị trí mã. Một vị trí mã nhị phân còn được gọi là vị trí nhị phân hay một bit. Một vị trí mã thập phân còn được gọi là một vị trí thập phân hay một digit. Chỉ số vị trí là số hiệu của vị trí mã trong từ mã theo một cách đánh số cụ thể. Hiện nay ta thường đánh số vị trí mã bên phải nhất của từ mã là vị trí 0 và sau đó cứ dịch sang trái một vị trí thì chỉ số vị trí tăng thêm một.

- Trọng số vị trí ω_i *(i=1..n)* là một hệ số nhân làm thay đổi giá trị của ký hiệu mã khi nó nằm ở các vị trí khác nhau. Trọng số vị trí này phụ thuộc vào mỗi mã hiệu cụ thể. Trong trường hợp số, trọng số vị trí là lũy thừa bậc chỉ số vị trí của cơ số của mã.

- Trọng số của từ mã *b*: là tổng các giá trị của các ký hiệu mã có trong từ mã.

$$b = \sum_{k=0}^{n-1} a_k \omega_k \tag{4.2}$$

trong đó a_k là giá trị riêng của ký hiệu mã ở vị trí k. Trong trường hợp mã hiệu là hệ đếm thì trọng số của từ mã là: $b = \sum_{k=0}^{n-1} a_k m^k$

- Khoảng cách mã *D*: Khoảng cách mã là khoảng cách giữa hai trọng số của hai từ mã tùy vào việc định nghĩa giá trị riêng và trọng số vị trí của các ký hiệu mã, ta sẽ có những định nghĩa khác nhau cho khoảng cách mã.

4.1.2 Điều kiện thiết lập bộ mã

Ta đã nói trong phần trên là mỗi từ mã là một tổ hợp mã dùng để mã hóa một tin hay một khối tin của nguồn. Vấn đề là có điều kiện nào ràng buộc để một tổ hợp mã sẽ được hay không được dùng làm một từ mã. Những điều kiện này sẽ được gọi là điều kiện thiết lập mã, chúng sẽ thể hiện như thế nào và kiểm tra chúng như thế nào là vấn đề chúng ta sẽ phải xác định trong phần này. Trước hết ta sẽ phát biểu những điều kiện thiết lập mã và sau đó sẽ đi tìm thể hiện của chúng và cách kiểm tra chúng.

- ***Điều kiện chung***

Các tin truyền đi được mã hoá thành dãy các ký hiệu liên tiếp, khi nhận tin ta phải giải mã được để thu được thông tin. Muốn vậy các ký hiệu phải được sắp xếp theo quy luật nào đó để tách đúng thông tin ban đầu.
Ví dụ: ta có 4 tin a, b, c, d được mã hoá bằng bộ mã nhị phân như sau:

a=00
b=01
c=10
d=11

Một tin ‘aaabcdb’ được mã hoá như sau: ‘00000001101101’ và truyền đi .

Khi nhận được tin và giải mã, nếu xác định được gốc của dãy ký hiệu trên, chúng ta chỉ có thể tách một cách duy nhất thành dãy tin ban đầu bằng cách từ gốc trở đi chia thành từng nhóm hai ký hiệu mã tương ứng. Như vậy bộ mã trên cho phép phân tích các từ mã một cách duy nhất và được gọi là mã phân tách được.

Cũng tin trên nếu ta mã bằng bộ mã khác:

a=0
b=01
c=101
d=1

Vẫn nguồn trên mã theo bộ mã này ta được: ‘00001101101’. Khi nhận tin và giải mã ta có thể giải mã như sau: ‘aaaaddbdb’ hoặc ‘aaabcdb’. Như vậy tin nhận được sẽ sai lạc so với nguồn vì vậy bộ mã này không dùng được. Vậy khái niệm mã phân tách được định nghĩa như sau:

Sự tồn tại quy luật cho phép tách được một cách duy nhất dãy các ký hiệu mã thành các từ mã được gọi là điều kiện thiết lập mã chung cho bộ mã. Bộ mã thỏa mãn điều kiện thiết lập mã còn được gọi là bộ mã phân tách được.

- ***Điều kiện riêng cho từng loai mã (mã đều và không đều)***

Đối với mỗi bộ mã còn tồn tại những điều kiện riêng phải được thỏa mãn khi thiết lập nó.

- Với mã không đều (mã thống kê tối ưu) ta phải chọn bộ mã sao cho đạt được độ dài trung bình mã tối thiểu.

- Với mã đều (mã sửa sai) thì bộ mã có khả năng phát hiện và sửa sai càng nhiều càng tốt.

Các điều kiện riêng cho mỗi bộ mã chính là những điều kiện về hình thức, về yêu cầu kỹ thuật hoặc chỉ tiêu kỹ thuật riêng mà bộ mã cần đạt được. Các điều kiện này là khác nhau với mỗi loại mã cụ thể.

4.2 Các phương pháp biểu diễn mã

4.2.1 Biểu diễn bằng bảng liệt kê (Bảng đối chiếu mã)

Đây là cách trình bày bộ mã đơn giản nhất. Người ta dùng bảng liệt kê những tin của nguồn và mã tương ứng của nó, bảng liệt kê có ưu điểm là cho thấy cụ thể tức thời tin và từ mã nhưng có nhược điểm là cồng kềnh và không cho thấy tầm quan trọng khác nhau của từng từ mã.

Ví dụ: Biểu diễn mã bằng bảng như sau:

Tin	a_1	a_2	a_3	a_4	a_5
Từ mã	00	01	100	1010	1011

4.2.2 Biểu diễn bằng tọa độ mã

Mỗi từ mã có hai thông số có thể xác định duy nhất mà không bị nhầm lẫn giữa các từ mã với nhau đó là độ dài *n* và trọng số *b*, nghĩa là không tồn tại hai từ mã bằng nhau đồng thời cả độ dài *n* và trọng số *b*.

Mặt tọa độ mã là một biểu diễn dựa trên hai thông số của từ mã là độ dài từ mã *n* và trọng số *b* để lập một mặt phẳng có hai tọa độ, trên đó mỗi từ mã được biểu diễn bằng một điểm. Trọng số *b* của từ mã là tổng trọng số các ký hiệu trong từ mã. Trọng số *b* được tính theo công thức:

$$b = \sum_{k=0}^{n-1} a_k m^k \qquad (4.3)$$

k: Là số thứ tự của ký hiệu thứ *k* trong từ mã

a_k: Là trị của ký hiệu thứ *k* (ví dụ mã nhị phân có hai trị là 0 và 1)

m: Là cơ số mã (mã nhị phân *m* = 2)

Ví dụ 1: Tính trọng số của các từ mã nhị phân sau:

Từ mã 1011 ta có $b = 1.2^0 + 1.2^1 + 0.2^2 + 1.2^3 = 11$

Từ mã 011 ta có $b = 1.2^0 + 1.2^1 + 0.2^2 = 3$

Ví dụ 2 : Cho các từ mã sau :

$a_1 = 00$ $n_1 = 2$ $b_1 = 0$

$a_2 = 10$ $n_2 = 2$ $b_2 = 2$

$a_3 = 100$ $n_3 = 0$ $b_3 = 4$

$a_4 = 101$ $n_4 = 3$ $b_4 = 5$

• **Định lý.** *Không có hai từ mã mã hóa hai tin khác nhau của cùng một bộ mã thỏa mãn đồng thời* $n_i = n_j$ *và* $b_i = b_j$ *(* $i \neq j$ *)*

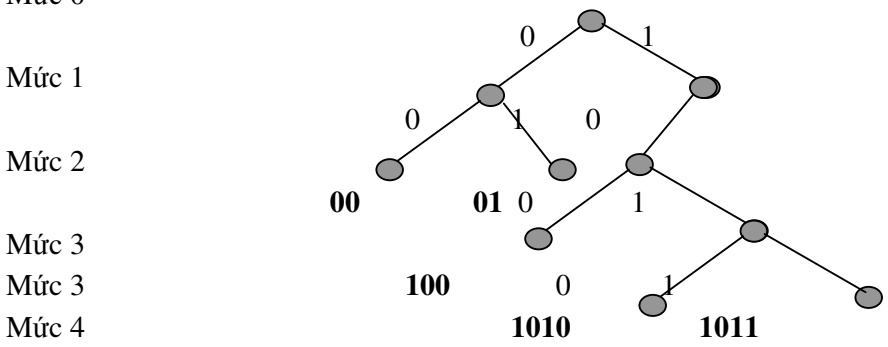
4.2.3 Đồ hình mã

Các phương pháp đồ hình sử dụng một đồ hình để biểu diễn một mã hiệu, nó cho phép trình bày mã một cách gọn hơn các bảng mã đồng thời cho thấy rõ các tính chất quan trọng của mã hiệu một cách trực quan hơn. Các phương pháp biểu diễn mã hiệu bằng đồ hình gồm có cây mã và đồ hình kết cấu.

• ***Biểu diễn bằng cây mã***

Cây mã là một đồ thị hình cây biểu diễn mã có các nút và nhánh, cây có một nút gốc duy nhất từ một nút có nhiều nhất là m nhánh (m là cơ số mã) mỗi nhánh là trị của ký hiệu, mỗi nhánh kết thúc ở một nút cao hơn. Nút cuối là đặc trưng cho một từ mã hình thành từ các trị trên các nhánh. Các từ mã mức trên có tầm qua trọng cao hơn các từ mã mức dưới.

Ví dụ: Có cây mã nhị phân có 5 từ mã sau:



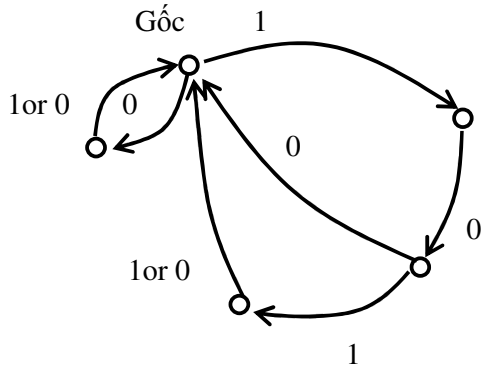
Hình 4.1

Trong ví dụ trên cho thấy khi nhìn vào cây mã ta biết cây mã có phải là cây mã đồng đều hay không đồng đều, loại mã đầy hay vơi.

• ***Đồ hình kết cấu***

Phương pháp này ta dùng một đồ thị có hướng gồm các nút và các nhánh, mỗi nhánh là một cung có hướng, mỗi từ mã là một chu trình theo chiều của cung đi từ gốc. Phương pháp này biểu diễn từ mã gọn nhẹ và trực quan

Ví dụ: Biểu diễn bộ mã nhị phân bằng đồ thị:



Hình 4.2

Sơ đồ này ta có 5 từ mã sau: 00, 01, 100, 1010, 1011

Đồ hình kết cấu không những dùng để mô tả bản thân từ mã mà còn dùng để xét cách vận hành thiết bị mã hóa và giải mã như là một đồ thị mô tả trạng thái của thiết bị.

4.2.4 Phương pháp hàm cấu trúc mã

Phương pháp này nói lên một đặc tính quan trọng của mã là sự phân bố các từ mã có độ dài khác nhau, ký hiệu bằng $G(n_i)$: số từ mã có độ dài là n_i. Từ hàm cấu trúc có thể phân biệt được mã đều hoặc không đều. Cũng từ hàm cấu trúc ta hoàn toàn có thể xác định được bộ mã có thỏa mãn điều kiện phân tách được hay không.

4.3 Mã có tính phân tách được, mã có tính prefix

Trong mục này ta xét các tiêu chuẩn được sử dụng để đánh giá một mã hiệu có thỏa mãn điều kiện thiết lập mã hay không. Ta biết rằng điều kiện chung để thiết lập mã là mã phải phân tách được cho nên tiêu chuẩn thiết lập mã chính là tiêu chuẩn để mã phân tách được hay chính là những tiêu chuẩn cho phép tách đúng từng từ mã từ chuỗi mã nhận được.

Lưu ý giữa từ mã và tin được mã hóa có quan hệ 1-1 thì việc giải mã ở phía thu sẽ bao gồm việc tách đúng từ mã và chuyển ngược từ mã thành tin tương ứng.

Việc chuyển từ mã thành tin được thực hiện nhờ một sơ đồ giải mã xác định. Việc tách đúng các từ mã là một thuật toán kiểm tra tính đúng của một số tiêu chuẩn được gọi là điều kiện phân tách của mã hiệu, việc kiểm tra này sẽ bắt đầu từ ký hiệu mã đầu tiên của chuỗi cho đến khi có thể cắt được một từ mã thì nó sẽ cắt từ mã và lại coi ký hiệu tiếp sau làm ký hiệu đầu tiên của chuỗi để kiểm tra tiếp.

Một trong những cách tiếp cận cơ bản nhất là trong bảng mã, hãy chọn 1 từ mã trùng với phần đầu của xâu mã sau đó xóa phần đầu của xâu mã và gộp kí hiệu tương ứng vào xâu gốc, quá trình sẽ dừng khi xâu mã đó bị xóa hết.

Thuật toán giải mã có thể mô phỏng như sau

```
Procedure Giai_Ma;
Input   st:string;{Xau da ma hoa}
        x:array[1..N] of char;{Bang ki hieu}
        b:array[1..N] of string;{Bang ma tuong ung}
Output  xaugoc:string;{xau goc ban dau}
BEGIN
        xaugoc:='';
        while length(st)>0 do
        for i:=1 to N do
        if b[i]=copy(st,1,lenght(b[i])) then
        begin
                xaugoc:=xaugoc+x[i];
                delete(st,1,lenght(b[i]));
        end;
END;
```

4.3.1 Điều kiện để mã phân tách được

Ta thấy khi nhận được một dãy ký hiệu mã để có thể phân tách từ mã một cách duy nhất và đúng đắn bộ mã phải thoả mãn điều kiện cần và đủ là:

" *Bất kỳ dãy từ mã nào của bộ mã cũng không được trùng vời một dãy từ mã khác*".

Ví dụ:

Cho bộ mã có các từ mã sau: 00 01 11 100 1010 1011

Nếu nhận được dãy: ‘1000101001011101101’
Ta chỉ có thể tách được duy nhất thành: ‘100-01-01-00-1011-1011-01’.
Như vậy bộ mã trên là loại bộ mã phân tách được

Khái niệm độ chậm giải mã là số ký hiệu nhận được cần thiết phải có mới phân tách được một từ mã. Độ chậm giải mã có thể là hữu hạn nhưng cũng có thể là vô hạn. Để xác định tính phân tách được của một bộ mã và độ chậm giải mã hữu hạn hay vô hạn ta xây dựng bảng thử như sau:

Bước 1: Sắp xếp các từ mã vào cột đầu tiên của bảng (cột 1)

Bước 2: So sánh các từ mã ngắn với các từ mã dài hơn trong cột 1, nếu từ mã ngắn giống phần đầu từ mã dài thì ghi phần còn lại trong từ mã dài sang cột 2.

Bước 3: Đối chiếu các tổ hợp mã trong cột 2 với các từ mã trong cột 1 lấy phần còn lại ghi vào cột tiếp theo (cột 3).

Bước 4: Đối chiếu các tổ hợp mã trong cột 3 với các từ mã trong cột 1… thực hiện giống như trên cho đến khi có một cột trống thì dừng.

Điều kiện cần và đủ để mã có thể phân tách là trong cột $j \geq 2$ không có tổ hợp nào trùng với một từ mã trong cột 1.

Để rõ hơn về thuật toán ta quan sát các ví dụ sau:

Ví dụ 1:

1	2	3
00	-	-
01	-	-
100	-	-
1010	-	-
1011	-	-

Bảng thử có các cột từ thứ 2 là rỗng nên bộ mã này phân tách được, độ chậm giải mã của bộ mã này bằng độ dài từ mã.

Như vậy có thể nói cách khác là để có tính phân tách được điều kiện cần và đủ là bất kỳ từ mã nào cũng không được trùng với phần đầu của từ mã khác trong cùng bộ mã.

Ví dụ 2:

1	2	3	4	5
10	0	1	0	…
100	1	11	00	…
01	-	0	1	…
011	-	00	11	…

Trong các cột từ 2,3,.. của bảng thử này không có tổ hợp mã nào trùng với các từ mã trong cột 1, nhưng có thể điền các cột j đến vô hạn mà không gặp cột trống. Bộ mã này phân tách được nhưng vì độ chậm giải mã là vô hạn nên trong trường hợp này có thể coi bộ mã là không phân tách được. Độ chậm giải mã được tính theo công thức sau: $\left[\frac{j-1}{2}\right] n_{\min} \le T_{ch} \le \left[\frac{j}{2}\right] n_{max}$. Trong đó j là số hiệu cột rỗng; $n_{\min}, n_{max}$ tương ứng là độ dài từ mã ngắn nhất và độ dài từ mã dài nhất.

4.3.2 Mã có tính prefix

Trong các loại mã có tính phân tách được, loại mã mang nhiều đặc điểm có ích cho việc khai thác sử dụng và được nghiên cứu nhiều là mã có tính prefix.

Prefix của một tổ hợp mã là bộ phận của tổ hợp mã đó sau khi bỏ đi một hay nhiều ký hiệu từ bên phải

Ví dụ 1: Cho tổ hợp mã: 01100 1110

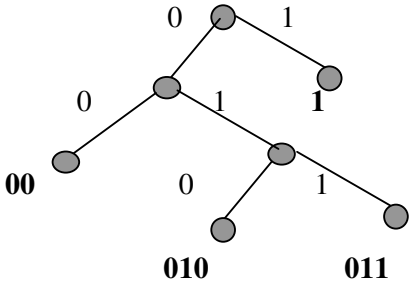
Có các prefix sau:

01100111
0110011
011001
01100
0110
011
01
0

Mã có tính prefix được định nghĩa như sau: *Một bộ mã được gọi là mã có tính prefix nếu bất kỳ từ mã nào cũng không phải là phần đầu của bất kỳ một từ mã khác trong bộ mã.*

Khi biểu diễn bằng cây mã, ta nhận thấy bộ mã có tính *prefix* khi các từ mã chỉ là nút lá. Mã đầy là bộ mã có tính *prefix*.

Ví dụ 2: Cây mã sau biểu diễn một bộ mã prefix



Hình 4.3

4.3.3 Bất đẳng thức Kraft

Để đưa ra được điều kiện tổng quát về tính phân tách được của từ mã, ta có nhận xét sau đối với mã có tính prefix với cơ số mã là m:

$$G(1) \le m$$
$$G(2) \le m^2 - mG(1)$$
$$G(3) \le m^3 - m^2G(1) - mG(2)$$
$$.....$$
$$G(n) \le m^n - m^{n-1}G(1) - ..... - mG(n-1)$$

Vì vậy: $\sum_{j=1}^{n} \frac{G(j)}{m^j} \le 1$ hay có thể viết tương đương như sau:

$\sum_{i=1}^{N} \frac{1}{m^{n_i}} \le 1$. Bất đẳng thức này được gọi là Bất đẳng thức Kraft. Trong đó N là số từ mã tương ứng với số tin của nguồn, n_i là độ dài từ mã mã hóa tin u_i.

Ngược lại một dãy số nguyên $n_1, n_2, ..., n_N$ thỏa mãn Bất đẳng thức Kraft thì sẽ tồn tại bộ mã có tính prefix nhờ thủ tục tạo mã prefix, điều này có thể mở rộng cho tất cả các loại mã phân tách được có hoặc không có tính prefix.

Thủ tục tạo mã prefix

Bước 1: Sắp xếp các từ mã theo thứ tự tăng dần của từ mã: $n_1 \le n_2 \le ... \le n_N$; xây dựng cây đầy đủ, mỗi nút có m nhánh, độ cao là $n_N + 1$.

Bước 2: Ở mức n_1 chọn một nút bất kỳ, và gán mã là từ mã C_1 và xóa các nút kề sau nó.

Bước 3: Lặp lại Bước 2 đối với mức $n_2, n_3, ..., n_N$ ta được các từ mã $C_1, C_2, ..., C_N$.

4.4 Mã thống kê tối ưu

4.4.1 Giới hạn độ dài trung bình của từ mã

- ***Giới hạn dưới***

Ta phải xác định độ dài trung bình của từ mã để đạt được tiêu chuẩn tối ưu như đã phân tích ở trên. Giả sử có nguồn tin $U = \{u_1, u_2, ..., u_N\}$ với các xác suất tương ứng $p(u_i)$ lượng tin trung bình là $H(U)$.

Ta chọn bộ mã X có cơ số m sao cho các xác suất $p(x_i)$ xấp xỉ bằng nhau, khi đó lượng tin của một ký hiệu mã là :

$$I(X) = \log m$$

Trong trường hợp mã nhị phân thì ta có $I(X) = 1$ (bit/k.h)

Nếu từ mã có độ dài n_i thì lượng tin chứa trong từ mã sẽ là $n_i \log m$. Nếu các từ mã có độ dài không bằng nhau thì lượng tin sẽ bằng $\bar{n} \log m$ trong đó $\bar{n}$ là độ dài trung bình của từ mã.

Mã hoá nguồn U trên với bộ mã X . Để đảm bảo không bị mất mát thông tin thì

$$n_i \log m \ge I(u_i) \Leftrightarrow n_i \ge \frac{\log \frac{1}{p(u_i)}}{\log m} \Rightarrow \sum_{i=1}^{N} n_i p(u_i) \ge \frac{1}{\log m} \sum_{i=1}^{N} p(u_i) \log \frac{1}{p(u_i)}$$

Vì vậy : $\bar{n} \log m \ge H(U) \Rightarrow \bar{n} \ge \frac{H(U)}{\log m}$

Như vậy độ dài trung bình của từ mã không nhỏ hơn tỉ số entropi của nguồn và lượng tin trung bình cực đại của một ký hiệu mã.

Với mã nhị phân ta có $\bar{n} \geq$ H(U), đó là giới hạn dưới của độ dài trung bình của từ mã. Dấu '=' xảy ra khi $n_i = I(u_i)$. Nếu mã hóa tin bằng một mã nhị phân có chiều dài n_i thì lượng tin chứa trong từ mã sẽ là n_i bit. Nếu lấy độ dài trung bình từ mã thì ta sẽ được lượng tin trung bình chứa trong từ mã. Thông thường $I(u_i)$ không phải là số nguyên nên điều kiện trên chỉ là điều kiện giới hạn mà dựa vào đó có thể xây dựng được các thuật toán xác định mã thống kê tối ưu.

- **Giới hạn trên**

Để thỏa mãn tiêu chuẩn của mã thống kê tối ưu thì đồ dài trung bình phải có giới hạn sau:

$$\frac{H(U)}{\log m} \leq \bar{n} \leq \frac{H(U)}{\log m} + 1 \qquad (4.4)$$

Mã nhị phân thì: $H(U) \leq \bar{n} \leq H(U) + 1$.

Như vậy có thể xây dựng bộ mã với độ dài trung bình không lớn hơn tỉ số entropi của nguồn với lượng tin trung bình cực đại chứa trong một ký hiệu cộng 1 đơn vị.

Tóm lại bộ mã có độ dài trung bình $\bar{n}$ thoả mãn các điều kiện trên gọi là mã thống kê tối ưu. Một bộ mã như vậy phải thoả mãn những đặc điểm sau:

- Các ký hiệu phải có cùng xác suất.
- Sự xuất hiện của các ký hiệu trong từ mã là độc lập với nhau tức là xác suất xuất hiện của ký hiệu sau không phụ thuộc vào sự có mặt của các ký hiệu ra trước.

4.4.2 Tiêu chuẩn mã thống kê tối ưu

Như đã đề cập ở phần trước về chiều dài trung bình của từ mã, tiêu chuẩn của mã thống kê tối ưu là đạt đến chiều dài trung bình của từ mã tối thiểu. Đây là một hướng lớn của mã hóa (mã nén dữ liệu). Do các tin của nguồn tin có xác suất xuất hiện khác nhau, nên việc dùng các từ mã có độ dài nhỏ để mã hóa cho các tin có xác suất xuất hiện cao sẽ làm cho số ký hiệu cần

thiết để mã hóa cho một chuỗi các tin nhỏ hơn và tính kinh tế cao hơn. Tính kinh tế của bộ mã được đo bằng công thức

$$\rho = \frac{H(U)}{\overline{n}\log m}) \tag{4.5}$$

Vậy nguyên tắc cơ bản của mã thống kê tối ưu dựa trên cơ sở: độ dài từ mã n_i tỷ lệ nghịch với xác suất xuất hiện $p(u_i)$ nghĩa là các từ mã dài sẽ dùng để mã hóa cho các tin có xác suất xuất hiện nhỏ và ngược lại.

4.4.3 Mã thống kê Fano –Shanon

Fano và Shanon đã độc lập nghiên cứu và đề xuất phương pháp xây dựng bộ mã thống kê tối ưu, bản chất của hai phương pháp là tương đương nhau. Để thuận tiện trong việc trình bày, các bộ mã xây dựng được từ các thuật toán này có cơ số mã $m = 2$

- ***Phương pháp mã theo Fano***

Giả sử có nguồn tin $U = \{u_1, u_2, ..., u_N\}$ với các xác suất tương ứng $p(u_i)$ $i = 1,2,...,N$

u_k	u_1	u_2	u_3	···	u_N
p_k	p_1	p_2	p_3	···	p_N

Nhà toán học Fano đề xuất thuật toán mã hoá như sau:

Bước 1: Sắp xếp các tin u_i theo thứ tự giảm dần của xác suất.

Bước 2: Chia các tin làm hai nhóm có xác suất xấp xỉ bằng nhau. Nhóm đầu lấy trị 0, nhóm sau lấy trị 1.

Bước 3: Lặp lại bước 2 đối với các nhóm con cho tới khi tất cả các nhóm chỉ còn lại một tin thì kết thúc thuật toán.

Để rõ hơn về thuật toán, ta xét ví dụ sau:

Cho nguồn tin U gồm 7 tin: $U = \{u_1, u_2, ..., u_7\}$

u_i	$p(u_i)$	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Từ mã
u_1	0,34	0	0				00
u_2	0,23	0	1				01

u_3	0,19	1	0				10
u_4	0,10	1	1	0			110
u_5	0,07	1	1	1	0		1110
u_6	0,06	1	1	1	1	0	11110
u_7	0,01	1	1	1	1	1	11111

Độ dài trung bình của từ mã:

$\bar{n} = 0{,}01.5 + 0{,}06\ .5 + 0{,}07.4 + 0{,}10.\ 3 + 0{,}19.\ 2 + 0{,}23.\ 2 + 0{,}34\ .2 = 2{,}41$

Trị số kinh tế $\rho = \frac{H(U)}{\bar{n}}$ = 2,37 /2,41 = 0,98.

Nhận xét:

1. Việc sắp xếp nguồn theo xác suất giảm dần nhằm mục đích đẩy các tin có xác suất cao lên đầu bảng cùng với việc chia đôi nguồn dẫn tới các lớp trên sẽ kết thúc rất nhanh, vì vậy các tin có xác suất cao sẽ có độ dài từ mã ngắn dẫn tới độ dài trung bình của bộ mã là nhỏ.

2. Độ phức tạp của thuật toán phụ thuộc vào việc sử dụng thuật toán sắp xếp. Nếu sử dụng thuật toán sắp xếp đệ quy thì độ phức tạp sẽ là $O(n\log n)$.

3. Xuất phát từ thuật toán Fano, ta có thể mở rộng cho việc tạo bộ mã với cơ số m bất kỳ bằng cách trong bước 2 ta chia thành m lớp và lấy các trị từ 0 đến $m-1$

4. Trong trường hợp khi có nhiều tin với xác suất bằng nhau cũng như có nhiều phương án phân lớp thì bộ mã thu được có thể không duy nhất.

Thuật toán trên được mô phỏng bởi chương trình sau:

```
Program Fano;
uses wincrt;
var    st:string;
         A:array[1..255] of char;
         Ma:array[1..255] of string[20];
         P:array[1..255] of real;;
         n,i,j,k:integer;
Procedure Input;
```

```
Begin
        write('cho xau can ma hoa:');readln(St);
end;
Procedure Output;
var k:integer;xauma:string;
Begin
      xauma:='';
      for i:=1 to length(st) do
      for k:=1 to n do
      if st[i]=a[k] then xauma:=xauma+ma[k]+' ';
      writeln('Ket qua ma hoa');
      writeln(xauma);
end;
Procedure Create;
var s:set of char;
Begin
        n:=0;s:=[];
      for i:=1 to length(st) do
        if not (St[i] in S) then
        Begin
                n:=n+1;
                A[n]:= St[i];
                S:=S+[St[i]];
        end;
for i:=1 to n do P[i]:=0;
for i:=1 to n do
begin
        for k:=1 to length(St) do
        if A[i]=St[k] then P[i]:= p[i]+1;
        P[i]:=P[i]/length(st);
end;
for i:=1 to n do Ma[i]:= '';
end;
```

```
Procedure Sorting;
var i,j,tgp:integer;
TgA: char;
Begin
        For i:=1 to n-1 do
        For j:=i+1 to n do
        If P[i]<P[j] then
        Begin
                TgA:=A[i]; A[i]:=A[j]; A[j]:=TgA;
                TgP:=P[i]; P[i]:=P[j]; P[j]:=TgP;
        End;
end;
Procedure Mahoa_Fano(l,r:integer);
Var k:integer;
   s,Sum:integer;
Begin
 if l<r then
  Begin
        S:=0;
       For i:=l to r do S:=S+P[i];
        Sum:=0; k:=l-1;
        Repeat
          k:=k+1;
          Sum:=Sum+P[k];
        until Sum>=S/2;
        For i:=l to K do Ma[i]:=Ma[i]+'0';
        For i:=K+1 to r do Ma[i]:=Ma[i]+'1';
        Mahoa_fano(l,k); Mahoa_fano(k+1,r);
  End;
End;
BEGIN
Input;
create;
```

Sorting;
Mahoa_fano(1,n);
Output;
Readln;
END.

- ***Phương pháp mã theo Shanon***

Xét nguồn U với bảng phân phối xác suất:

u_k	u_1	u_2	u_3	...	u_N
p_k	p_1	p_2	p_3	...	p_N

Nhà toán học Shanon đề xuất thuật toán mã hóa như sau:

Bước 1: Sắp xếp các tin u_i theo thứ tự giảm dần của xác suất.

Bước 2: Tính $F(u_i)$ theo công thức:

$$F(u_i) = \sum_{j=0}^{i-1} p(u_j) \qquad \text{nếu } i \geq 2$$

$$F(u_i) = 0 \qquad \text{nếu } i = 1$$

Ở bước này ta đã giả thiết các tin được sắp xếp theo thứ tự giảm dần của xác suất: $p(u_1) \geq p(u_2) \geq ... \geq p(u_N)$

Bước 3: Tính độ dài từ mã mã hoá tin u_i theo công thức sau:

$$I(u_i) \leq n_i \leq I(u_i) + 1 \text{ hoặc } 2^{-n_i} \leq p(u_i) \leq 2^{1-n_i}$$

Bước 4 : Chuyển đổi $F(u_i)$ tính được ở Bước 2 từ hệ thập phân sang hệ nhị phân.

Bước 5 : Xác định từ mã (n_i, b_i) bằng cách lấy n_i ký hiệu nhị phân ngay sau dấu phẩy.

Để rõ hơn về thuật toán ta xét ví dụ sau :

Cho nguồn tin U gồm 7 tin: $U = \{u_1, u_2, ..., u_7\}$

u_i	u_1	u_2	u_3	u_4	u_5	u_6	u_7
$p(u_i)$	0,34	0,23	0,19	0,10	0,07	0,06	0,01

Thực hiện qua 5 bước theo thuật toán ta được kết quả sau:

u_i	$p(u_i)$	n_i	$F(u_i)$	$F(u_i)$ hệ 2	Từ mã
u_1	0,34	2	0,0	0,00	00
u_2	0,23	3	0,34	0,010	010
u_3	0,19	3	0,57	0,100	100
u_4	0,10	4	0,76	0,1100	1100
u_5	0,07	4	0,86	0,1101	1101
u_6	0,06	5	0,93	0,11101	11101
u_7	0,01	7	0,99	0,1111110	1111110

Kiểm tra tính tối ưu:

$$H(U) = -\sum_{i=1}^{7} p(u_i) \log p(u_i)$$

= -(0,01 log 0,01 + 0,06log0,06 + 0,10log0,10 + 0,19log0,19 + 0,23log0,23 + 0,34log0,34) ≈ 2,37

$\bar{n} = \sum_{i=1}^{7} n_i p(u_i) = 0{,}01.7 + 0{,}06.5 + 0{,}07.4 + 0{,}10.4 + 0{,}19.3 + 0{,}23.3 + 0{,}34.2 =$ 2,99.

Trị số kinh tế $\rho = H(U) / \bar{n} = 2{,}37/2{,}99 = 0{,}81$.

Nhận xét:

1. Việc sắp xếp nguồn theo xác suất giảm dần cũng nhằm mục đích dẫn tới độ dài trung bình của bộ mã là nhỏ.

2. Độ phức tạp của thuật toán phụ thuộc vào việc sử dụng thuật toán sắp xếp. Nếu sử dụng thuật toán sắp xếp đệ quy thì độ phức tạp sẽ là $O(n \log n)$.

3. Xuất phát từ thuật toán Shanon, ta có thể mở rộng cho việc tạo bộ mã với cơ số m bất kỳ bằng cách xác định lại độ dài n_i cũng như đổi các xác suất phụ sang dạng m phân.

4. Trong trường hợp khi có nhiều tin với xác suất bằng nhau thì bộ mã thu được có thể không duy nhất.

Thuật toán trên được mô phỏng bởi chương trình sau

```
Program shanon;
uses wincrt;
var     st:string;
          A:array[1..255] of char;
          Ma:array[1..255] of string[20];
          P:array[1..255] of real;
          PP:array[1..255] of real;
          n,i,j,k:integer;
Procedure Input;
Begin
          write('cho xau can ma hoa:');readln(St);
end;
Procedure Output;
var k:integer;xauma:string;
Begin
       xauma:='';
       for i:=1 to length(st) do
       for k:=1 to n do
       if st[i]=a[k] then xauma:=xauma+ma[k];
       writeln('Ket qua ma hoa');
       writeln(xauma);
end;
Procedure Create;
var s:set of char;
Begin
         n:=0;s:=[];
         for i:=1 to length(st) do
```

```
        if not (St[i] in S) then
        Begin
                n:=n+1;
                A[n]:= St[i];
                S:=S+[St[i]];
        end;
        for i:=1 to n do P[i]:=0;
        for i:=1 to n do
        begin
                for k:=1 to length(St) do
                if A[i]=St[k] then P[i]:= p[i]+1;
                p[k]:=p[k]/length(st);
        end;
end;
Procedure Sorting;
var i,j:integer;tgp:real;
TgA: char;
Begin
        For i:=1 to n-1 do
        For j:=i+1 to n do
        If P[i]<P[j] then
        Begin
                TgA:=A[i]; A[i]:=A[j]; A[j]:=TgA;
                TgP:=P[i]; P[i]:=P[j]; P[j]:=TgP;
        End;
end;
Procedure Mahoa_Shanon;
Var k:integer;
    s:real;
    nn:array[1..255] of integer;
    Begin
    for i:=1 to n do ma[i]:='';
    for i:=1 to n do
```

```
    begin
        nn[i]:=0;S:=1;
        repeat nn[i]:=nn[i]+1;S:=S*0.5;until S<P[I];
    end;
    for i:=1 to n do
    begin
        pp[i]:=0;
        for j:=1 to i-1 do pp[i]:=pp[i]+p[j];
    end;
    for i:=1 to n do
    repeat
         pp[i]:=2*pp[i];
         if pp[i]>1 then
         begin
             ma[i]:=ma[i]+'1';pp[i]:=pp[i]-1;
         end
         else ma[i]:=ma[i]+'0';
    until length(ma[i])=nn[i];
End;
BEGIN
Input;
create;
Sorting;
Mahoa_shanon;
Output;
Readln;
END.
```

Có thể chứng minh rằng những bộ mã được xây dựng theo hai phương pháp trên đều là mã prefix và hai phương pháp trên là tương đương nhau.

• ***Mã Huffman***

Huffman đã đưa ra nguyên tắc để xây dựng bộ mã thống kê tối ưu như sau: để một mã prefix có độ dài tối thiểu điều kiện cần và đủ là thoả mãn 3 tính chất:

Tính chất 1: Tính thứ tự của độ dài từ mã: Nếu sắp xếp các tin theo thứ tự giảm dần của xác suất $p(u_i) \geq p(u_j)$ với i < j thì độ dài của các từ mã tương ứng phải thỏa mãn điều kiện $n_i \leq n_j$.

Tính chất 2: Tính chất những từ mã cuối: Với n_0 thoả mãn điều kiện $2 \leq n_0 \leq m$ (m là cơ số mã) thì n_0 từ mã cuối luôn có độ dài n bằng nhau, về trọng số chỉ khác nhau ký hiệu bên phải

$$n_{N-n0} = .. = n_{N-1} = n_N .$$

Tính chất 3: Tính liên hệ từ mã cuối và từ mã trước cuối: Một dãy gồm n_N -1 ký hiệu phải là một từ mã hay là prefix của từ mã thứ N là từ mã cuối cùng của bộ mã.

Từ những nguyên tắc trên ta có thuật toán xây dựng cây mã Huffman như sau :

Bước 1: Chọn n_0 là số nguyên lớn nhất thoả mãn điều kiện: $2 \leq n_0 \leq m$ và $\frac{N-1}{n_0-1}$ là một số nguyên.

Bước 2: Khởi tạo danh sách N cây cấp n_0 chứa các trọng số $p(u_1), p(u_2),..., p(u_N)$ cho các tin $u_1, u_2,..., u_N$.

Bước 3: *For i :=1 to* $\frac{N-n_0}{n_0-1}$ *do*

Begin

(3.1) Tìm n_0 cây có trọng số thấp nhất: $T_1^{(i)}, T_2^{(i)},..., T_{n_0}^{(i)}$ tương ứng với trọng số là $p_1^{(i)}, p_2^{(i)},..., p_{n_0}^{(i)}$. Giả sử $p_1^{(i)} \geq p_2^{(i)} \geq ... \geq p_{n_0}^{(i)}$.

(3.2) Thay thế n_0 cây này bằng cây $T^{(i)}$ có trọng số $p_1^{(i)} + p_2^{(i)} + ... + p_{n_0}^{(i)}$ và có n_0 cây con là $T_1^{(i)}, T_2^{(i)},..., T_{n_0}^{(i)}$.

(3.3) Gán các giá trị riêng $0,1,2,...,n_0-1$ cho các nhánh trên cây $T^{(i)}$ tương ứng các nhánh nối với các cây con $T_1^{(i)},T_2^{(i)},...,T_{n_0}^{(i)}$.

End;

Cây cuối cùng thu được là cây mã gồm các nút lá là các từ mã tương ứng với các tin của nguồn. Mỗi từ mã là dãy các ký hiệu mã $\in \{0,1,...,n_0-1\}$ được đánh dấu trên đường đi từ gốc tới nút lá.

Thuật toán mã hóa trên có thể mô tả một cách khác như sau:

Bước 1: Sắp xếp nguồn X theo xác suất $p(u_i)$ giảm dần

Bước 2: Chọn n_0 là số nguyên lớn nhất thoả mãn điều kiện: $2 \le n_0 \le m$ và $\frac{N-1}{n_0-1}$ là một số nguyên.

Bước 3: Lặp, thay thế n_0 tin cuối cùng có xác suất nhỏ nhất thành 1 tin phụ có xác suất bằng tổng xác suất trong nhóm sau đó sắp xếp các tin cũ lại cùng tin phụ theo xác suất giảm dần. Quá trình lặp lại cho đến khi tin phụ có xác suất bằng 1 (Số bước lặp là $N-1$).

Bước 4: Lập mã: Xuất phát từ tin phụ có xác suất bằng 1, đánh các tin trong nhóm các trị từ $0...n_0-1$, từ các tin phụ sau trở đi, mã của các tin trong nhóm chính bằng mã của tin phụ cộng với các trị từ $0...n_0-1$. Quá trình sẽ dừng lại khi tất cả các tin đó được gán mã.

Bước 5: Loại bỏ tất cả các tin phụ, ta thu được bộ mã cần tìm.

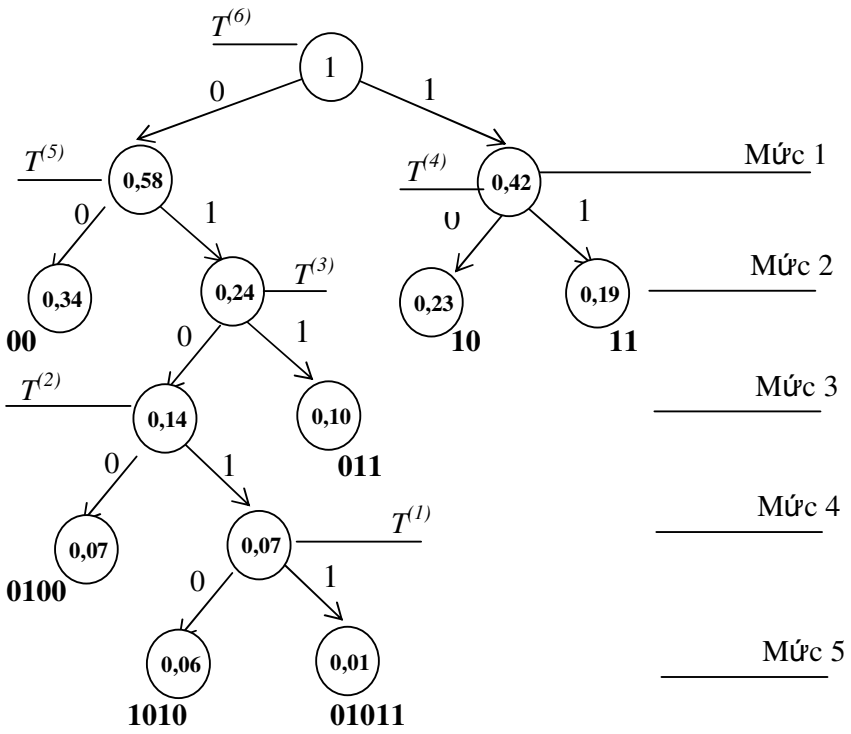
Để rõ hơn về thuật toán ta xét ví dụ sau:

Cho nguồn tin U gồm 7 tin: $U=\{u_1,u_2,....,u_7\}$ với cấu trúc thống kê

u_i	u_1	u_2	u_3	u_4	u_5	u_6	u_7
$p(u_i)$	0,34	0,23	0,19	0,10	0,07	0,06	0,01

Ta lấy cơ số mã *m=2*, nên chọn n_0=2

Sau khi thực hiện thuật toán ta được kết quả sau:



Thuật toán trên được mô phỏng bởi chương trình sau

```
Program huffman;
uses wincrt;
var     st:string;
          A:array[1..255] of char;
          Ma:array[1..255] of string[20];
         mma:array[1..255] of string[20];
          P:array[1..255] of real;
         aa:array[1..255] of char;
         pp:array[1..255] of real;
          n,i,j,k:integer;
Procedure Input;
Begin
          write('cho xau can ma hoa:');readln(St);
end;
```

```
Procedure Output;
var k:integer;xauma:string;
Begin
      xauma:='';
      for i:=1 to length(st) do
      for k:=1 to n do
      if st[i]=a[k] then xauma:=xauma+ma[k]+' ';
      writeln('Ket qua ma hoa');
      writeln(xauma);
end;
Procedure Create;
var s:set of char;
Begin
        n:=0;s:=[];
      for i:=1 to length(st) do
        if not (St[i] in S) then
        Begin
          n:=n+1;
              A[n]:= St[i];
              S:=S+[St[i]];
        end;
for i:=1 to n do P[i]:=0;
for i:=1 to n do
begin
        for k:=1 to length(St) do
        if A[i]=St[k] then P[i]:= p[i]+1;
        p[i]:=p[i]/length(st);
end;
for i:=1 to n do Ma[i]:= '';
end;
Procedure Sorting(n1:integer);
var i,j,tgp:integer;
TgA: char;
```

```
Begin
        For i:=1 to n1-1 do
        For j:=i+1 to n1 do
        If Pp[i]<Pp[j] then
        Begin
                TgA:=Aa[i]; Aa[i]:=Aa[j]; Aa[j]:=TgA;
                TgP:=Pp[i]; Pp[i]:=Pp[j]; Pp[j]:=TgP;
        End;
end;
Procedure Mahoa_huffman;
Var i,k:integer;
   b:array[1..255] of boolean;
Begin
    for i:=1 to 2*n-1 do
    begin
        mma[i]:='';b[i]:=true;
    end;
    for i:=1 to n do
    begin
        aa[i]:=a[i];pp[i]:=p[i];
    end;
    sorting(n);
    for k:=1 to n-1 do
    begin
        aa[n+k]:='#';
        pp[n+k]:=pp[n-k+1]+pp[n-k];
        sorting(n+k);
    end;
    mma[1]:='0';mma[2]:='1';
    b[1]:=false;b[2]:=false;
    i:=0;
    repeat
         i:=i+1;
```

```
        if aa[i]='#' then
        begin
           k:=i;
           repeat k:=k+1;until b[k]=true;
           mma[k]:=mma[i]+'0';mma[k+1]:=mma[i]+'1';
           b[k]:=false;b[k+1]:=false;
        end;
    until i=2*n-1;
    for i:=1 to n do
    for j:=1 to 2*n-1 do
    if a[i]=aa[j] then ma[i]:=mma[j];
End;
BEGIN
Input;
create;
Mahoa_huffman;
Output;
Readln;
END.
```

4.5 Thuật toán mã hoá Lempel-Ziv

Ta thấy rằng thuật toán mã hoá nguồn Huffman là thuật toán mã hoá nguồn tối ưu theo nghĩa bộ mã tạo ra có tính prefix và có độ dài trung bình tối thiểu. Để mã hoá cho nguồn rời rạc không nhớ, ta phải biết xác suất xuất hiện của tất cả các ký hiệu, và đối với nguồn có nhớ ta phải biết hàm mật độ xác suất của các khối ký hiệu. Tuy nhiên trong thực tế tính chất thống kê của nguồn thường không biết trước mà ta thường ước lượng các giá trị xác suất của nguồn rời rạc bằng cách quan sát một chuỗi dài các ký hiệu. Ngoại trừ việc ước lượng các xác suất p_k tương ứng với tần suất xuất hiện các ký hiệu riêng rẽ của nguồn, độ phức tạp tính toán trong việc ước lượng các xác suất đồng thời là rất cao. Như vậy sử dụng thuật toán mã hoá nguồn Huffman cho các nguồn có nhớ trong thực tế nói chung rất phức tạp. Trái lại với thuật toán

Huffman, thuật toán mã hoá Lempel-Ziv là độc lập với tính chất thống kê của nguồn. Trong thuật toán này gồm hai quá trình: Lập mã và giải mã.

Lập mã

Giả sử có nguồn tin gồm N ký hiệu: Bản tin u=$u_1u_2..u_n$ cần mã hoá bằng bộ mã có cơ số mã $m=2$. Tập ký hiệu mã M= {0,1}

Mã hóa dãy ký tự nguồn theo các bước sau:

Bước 1: Phân đoạn nguồn: $X=I_1I_2..I_m$.

$I_1 = u_1$, I_2 = đoạn ngắn nhất không trùng với I_1.

.. ..

Giả sử $I_1, I_2, .., I_k$ đã được xác định thì I_{k+1}: là đoạn ngắn nhất không trùng với $I_1, I_2, .., I_k$.

Ví dụ: 221022122011.

I_1=2; I_2=21; I_3=0; I_4=22; I_5=1; I_6=220; I_7=11

Bước 2: Biểu diễn dưới dạng cặp đôi (r_i , s_i) trong đó s_i là ký tự cuối cùng của từ phân đoạn I_i và r_i là số thứ tự của phân đoạn trước đó (số thứ tự của phân đoạn mà bỏ đi ký tự cuối cùng). Nếu phân đoạn chỉ có một số thì r_i=0

Ví dụ: (0, 2); (1,1); (0,0); (1, 2); (0,1); (4,0); (5,1)

Bước 3: Biểu diễn các cặp số (r_i, s_i) dưới dạng thập phân.

$$D_i = r_i *N + s_i.$$

$D_1 = 2$; $D_2 = 4$; $D_3 = 0$; $D_4 = 5$; $D_5 = 1$; $D_6 = 12$; $D_7 = 16$

trong đó: N = số phần tử của bộ ký hiệu nguồn.

Bước 4: Biểu diễn nhị phân các số vừa nhận dược

$B_1 = 10$; $B_2 = 100$; $B_3 = 0$; $B_4 = 101$; $B_5 = 1$; $B_6 = 1100$; $B_7 = 10000$.

Bước 5: Tính độ dài từ mã: phân đoạn thứ i có độ dài $l_i = \lceil \log(k * i) \rceil$

$l_1 = 2$; $l_2 = 3$; $l_3 = 4$; $l_4 = 4$; $l_5 = 4$; $l_6 = 5$; $l_7 = 5$

Bước 6: Gán thêm các số 0 vào bên trái biểu diễn nhị phân cho đủ độ dài l_i (nếu cần thiết). Dãy mã nhận được: 1010000000101000010110010000

Giải mã

Khi nhận được thông tin ở phía nhận cần giải mã. Giả sử dãy mã nhận được là y=$y_1y_2..y_n$.

Thuật toán:

Bước 1: Phân đoạn mã theo độ dài l_i đã biết.
Bước 2: Biểu diễn dưới dạng thập phân cho các phân đoạn mã
Bước 3: Biểu diễn dưới dạng cặp số (r_i, s_i).
Bước 4: Khôi phục dãy nguồn.
Ví dụ : Cho dãy mã 10100000001010001011001000 0
Tập ký hiệu nguồn {0,1,2}
Bước 1 : Phân đoạn dãy mã.
k=3
$l_1 = \lceil \log(3*1) \rceil = 2$, $l_2 = \lceil \log(3*2) \rceil = 3$, $l_3 = 4$, $l_4 = 4$, $l_5 = 4$, $l_6 = 5$, $l_7 = 5$
Vậy ta có : $B_1 = 10$; $B_2 = 100$; $B_3 = 0$; $B_4 = 101$; $B_5 = 1$; $B_6 = 1100$; $B_7 = 10000$.
Bước 2 : Đổi về thập phân cho các phân đoạn mã.
$D_1 = 2$; $D_2 = 4$; $D_3 = 0$; $D_4 = 5$; $D_5 = 1$; $D_6 = 12$; $D_7 = 16$
Bước 3 : Biểu diễn dưới dạng cặp số (r_i, s_i).
$(r_1, s_1) = (0,2)$; $(r_2, s_2) = (1,1)$; $(r_3, s_3) = (0,0)$; $(r_4, s_4) = (1,2)$; $(r_5, s_5) = (0,1)$;
$(r_6, s_6) = (4,0)$ $(r_7, s_7) = (5,1)$
Bước 4: Viết lại phân đoạn nguồn tương ứng với B_i.
$I_1 = 2$, $I_2 = 21$, $I_3 = 0$, $I_4 = 22$, $I_5 = 1$, $I_6 = 220$, $I_7 = 11$.
Vậy dãy nguồn đã cho là: u = 221022122011.

4.6 Mã chống nhiễu

Phương pháp mã hoá thống kê làm cho cấu trúc thống kê của nguồn trở nên hợp lý. Trong phần trước (tốc độ lập tin) đã có một ví dụ nói đến việc nâng cao entropi của nguồn bằng cách mã hoá hai lần, muốn như vậy cơ số của mã cuối cùng ít nhất cũng phải bằng số tin của nguồn cũ. Điều này không thuận tiện cho việc truyền tin. Thông thường người ta chỉ dùng các loại mã có cơ số bé (m=2 hoặc m=3). Vì vậy phương pháp mã hoá thống kê làm cho cấu trúc thống kê của nguồn tin trở nên hợp lý có nghĩa là làm cho entropi của bộ mã được dùng có trị số cực đại và độ dài trung bình của từ mã $\bar{n} = \dfrac{H}{\log m}$.

Sự áp dụng các loại mã thống kê trong các hệ thống truyền tin cho phép đạt được những chỉ tiêu kinh tế tốt, nghĩa là với lượng tin cần truyền đã cho thì

thời gian truyền tin sẽ được rút ngắn so với các phương pháp mã hoá khác, hoặc là khi thời gian truyền tin như nhau, lượng tin truyền đi sẽ được nâng lên.

Nhưng khi chú ý đến ảnh hưởng phá huỷ tin của nhiễu trong kênh, phương pháp mã hoá ngoài chỉ tiêu kinh tế nói trên còn cần phải đảm bảo chỉ tiêu truyền tin chính xác, nói cách khác chỉ tiêu chống nhiễu. Để giải quyết vấn đề này có hai xu hướng: một xu hướng xây dựng lý luận chống nhiễu của mã thống kê tối ưu (mã không đều) và xu hướng thứ hai là xây dựng cơ sở lý thuyết chống nhiễu các mã sửa sai và phát hiện sai.

4.6.1 Khái niệm về mã phát hiện sai và sửa sai

Như trên ta biết bộ mã đồng đều các từ mã có cùng độ dài n do vậy giữa các từ mã có sự sai nhầm, ta phải có cơ chế để phát hiện và sửa được những sai nhầm trong quá trình truyền tin. Bộ mã có tính chất phát hiện và sửa sai gọi là mã sửa sai

Bộ mã sửa sai gồm: Mã phát hiện sai và mã sửa sai

Các dạng sai nhầm của các từ mã tuỳ thuộc tính chất thống kê của kênh. Có thể phân loại như sau:

- Sai độc lập: Một hay nhiều ký hiệu sai nhầm nhưng các sai nhầm đó không ảnh hưởng lẫn nhau.

- Sai tương quan: Những ký hiệu sai phụ thuộc vào nhau thường xảy ra từng chùm liền nhau gọi là sai cụm.

Sự lựa chọn cấu trúc của mã chống nhiễu phải dựa trên tính chất thống kê của kênh, nói cách khác dựa trên sự phân bố xác suất sai nhầm trong kênh.

Ví dụ:

Hãy xác định quy luật phân bố xác suất sai nhầm trong một kênh nhị phân đối xứng. Kênh nhị phân đối xứng thuộc loại kênh chứa sai nhầm không tương quan và $p(1|0) = p(0|1) = p_s$.

Xác suất nhận đúng một ký hiệu sẽ là: $1-p_s$ và xác suất nhận đúng một từ mã có độ dài n là $(1-p_s)^n$, xác suất nhận sai từ mã đó là $p^n = 1-(1-p_s)^n$

Bây giờ ta xác định xác suất nhận sai t ký hiệu trong một từ mã gồm n ký hiệu. Trước tiên xác suất xảy ra *t* sai đồng thời: p_s^t và xác suất xảy ra trong một từ mã có *n* ký hiệu, *n-t* ký hiệu xuất hiện đúng: $(1-p_s)^{n-t}$, vậy xác suất xảy ra trong một từ mã *n* ký hiệu có một ký hiệu sai là: $p_1^{(n,t)} = p_s^t(1-p_s)^{n-t}$.

Tổng số các từ mã *n* ký hiệu có *t* ký hiệu sai là $C_n^t = n!/((n-t)!t!)$.
Xác suất xuất hiện một từ mã *n* ký hiệu có *t* ký hiệu sai là:

$$p(n,t) = C_n^t p_s^t (1-p_s)^{n-t}.$$

Khi trị số p_s nhỏ thì xác suất nhận các từ mã sai ít ký hiệu lớn hơn xác suất nhận các từ mã sai nhiều ký hiệu: $p(n,t) > p(n,t')$ nếu $t < t'$.
Điều này cho phép ta xây dựng các mã hiệu chống nhiễu hữu hiệu trong kênh nhị phân đối xứng. Các mã hiệu này có khả năng phát hiện và sửa sai các tổ hợp mã có số ký hiệu sai bé, nhiều hơn là đối với các tổ hợp mã có số ký hiệu sai lớn.

4.6.2 Cơ chế phát hiện sai

Nếu mã hiệu là tập hợp các từ mã *n* ký hiệu thì số từ mã được chọn phải bé hơn tổng số các tổ hợp có thể có từ *n* ký hiệu. Những tổ hợp không dùng làm từ mã gọi là tổ hợp cấm. Khi nhận tin nhận được tổ hợp cấm ta phát hiện ra sai. Với mã nhị phân có *n* ký hiệu thì có số tổ hợp là: $N_0 = 2^n$.

Mã muốn phát hiện sai được thì số tổ hợp dùng $N < N_0$ và ta có số tổ hợp cấm là: $N_0 - N$.

Ví dụ: $n = 2$ dùng 4 tổ hợp mã hoá tin a_1, a_2, a_3, a_4 như sau:

$$a_1 = 00, a_2 = 01, a_3 = 10, a_4 = 11$$

Khi nhận được bất kỳ tổ hợp nào ta cũng thấy hợp lý vì vậy ta không phát hiện ra sai nhầm. Nhưng nếu ta chỉ dùng 3 tổ hợp như sau: $a_1 = 00, a_2 = 01, a_3 = 10$ thì khi nhận được tổ hợp 11 ta biết là đã có sự sai nhầm. Tổ hợp 11 chính là tổ hợp cấm

Cơ chế sửa sai

Cơ chế sửa sai của mã hiệu đồng thời cũng là nguyên lý giải mã phải dựa vào tinh chất thống kê của kênh để đảm bảo mục tiêu sai nhầm tối thiểu. Muốn vậy cần phải dựa vào tính chất nhiễu trên kênh để phân nhóm các tổ hợp cấm, mỗi nhóm tương ứng với một từ mã mà chúng có khả năng bị chuyển đổi sang nhiều nhất.

Căn cứ vào đặc tính thống kê số từ mã sai ít ký hiệu xảy ra nhiều hơn nên ta ưu tiên xử lý trước. Coi mỗi từ mã là một véc tơ, mỗi ký hiệu là một

toạ độ của véc tơ. Nhiễu cũng được coi như là một véc tơ gọi là véc tơ sai. Từ mã sai coi như là tổng hợp của véc tơ mã và véc tơ sai.

Ví dụ 1:

Véc tơ mã: 0101, véc tơ sai: 0100, từ mã sai: 0001

Từ mã trên bị sai ký hiệu thứ 2 (bit 1 trở thành bit 0)

Ví dụ 2: Với một từ mã có 4 ký hiệu véc tơ sai cũng có 4 ký hiệu, có các phương án sai 1 ký hiệu, 2 ký hiệu, 3 ký hiệu .. Ta có bảng sau:

Vec tơ mã / Vec tơ sai	a_1 0001	a_2 0101	a_3 1110	a_4 1111	Số Kh sai
0001	0000	0100	-	-	1
0010	0011	0111	1100	1101	
0100	-	-	1010	1011	
1000	1001	1101	0110	0111	
0011	0010	0110	1101	1100	2
0101	0100	0000	1011	1010	
1001	1000	1100	0111	0110	
1010	1011	-	0100	-	
1100	1101	1001	0010	0011	
0111	0110	0010	1001	1000	3
1011	1010	-	-	0100	
1101	1100	1000	0011	0010	
1110	-	1011	0000	-	
1111	-	1010	-	0000	4

Với nhận xét trên ta có các tổ hợp cấm được phân thành nhóm, các từ mã cấm trong nhóm B_i tương ứng với từ mã đúng a_i. Ta được bảng sau:

B_1	B_2	B_3	B_4
0000	0100	1100	1101
0011	0111	1010	1011
1001	1101	0110	0111

Ta thấy nhận được tổ hợp ‘0111’ hoặc ‘1101’ do sai 1 ký hiệu ta có thể giải mã và sửa thành ‘0101’ (a_2), như vậy theo nguyên tắc trên ta có thể sửa sai cho bộ mã. Tuy nhiên như bộ mã trên cũng cho thấy có thể sửa không chính xác vì tổ hợp ‘1101’ có thể sửa thành a_2 cũng có thể sửa thành a_4. Để giải quyết người ta khắc phục bằng cách dùng bảng chọn hoặc tính khoảng cách mã của bộ mã.

4.6.3 Xây dựng bộ mã sửa sai bằng bảng chọn

Bằng cách lập bảng chọn ta chọn được tổ hợp từ mã dùng, hợp lý đảm bảo sửa sai được.

Ví dụ: Xâydựng bảng mã có 4 từ mã mỗi từ mã có 4 ký hiệu.

Xây dựng bảng có $2^n = 2^4$ cột như sau:

số cột	1	2	3	4	5	6	7		12	13	14	15	16
u \ *e*	0000	0001	0010	0011	0100	0101	0110	...	1011	1100	1101	1110	1111
0011	0011	0010	0001	0000	0111	0110	0101	...	1000	1111	1110	1101	1100
0110	0110	0111	0100	0101	0010	0011	0000	...	1101	1010	1001	1000	1001
1100	1100	1101	1110	1111	1000	1001	1010	...	0111	0000	0001	0010	0011

v: vec tơ mã, e: vec tơ sai.

Từ bảng trên ta chọn các từ mã để sử dụng bằng cách:

Chọn một từ bất kỳ ví dụ chọn từ mã $a_2 = 0001$ ta xem bản thân nó và các tổ hợp cấm của nó nếu trùng với các tổ hợp của từ mã a_i nào đó thì đánh dấu cột i để loại. Như trên loại cột 3, 5, 8, 12, 14, 15. Chọn từ mã thứ 2 trong các cột còn lại giả sử chọn a_6 loại các cột có các tổ hợp trùng với các tổ hợp cấm của a_6 ta sẽ loại cột 1, 4, 7, 10, 11, 16.. Chọn tiếp tương tự ta chọn được 4 từ mã là:

$a_1 = 0001$ (cột 2)

$a_2 = 0101$ (cột 6)

$a_3 = 1000$ (cột 9)

$a_4 = 1100$ (cột 13)

Với 4 từ mã trên khi nhận được mã sai ứng với kênh đã phân tích ở trên ta sẽ sửa sai tương đối chính xác ví dụ:

B1: 0010, 0111, 1101 sẽ sửa duy nhất thành a_1

B2: 0110, 0011, 1001 sẽ sửa duy nhất thành a_2

B3: 1011, 1110 , 0100 sẽ sửa duy nhất thành a_3

B4: 1111, 1010, 0000 sẽ sửa duy nhất thành a_4

4.6.4 Xây dựng bộ mã sửa sai bằng trọng số Hamming và khoảng cách Hamming

Khoảng cách Hamming được đặt theo tên của nhà toán học Richard Hamming, người giới thiệu lý thuyết này trong tài liệu có tính cơ sở của ông về mã phát hiện lỗi và sửa lỗi (*error-detecting and error-correcting codes*). Nó được sử dụng trong kỹ thuật viễn thông để tính số lượng các bit trong một từ nhị phân (*binary word*) bị đổi ngược như một hình thức để ước tính số lỗi xảy ra trong quá trình truyền thông, và vì thế đôi khi nó còn được gọi là khoảng cách tín hiệu (*signal distance*). Việc phân tích trọng lượng Hamming của các bit còn được sử dụng trong một số ngành bao gồm lý thuyết tin học, lý thuyết mã hóa, và mật mã học. Tuy vậy, khi so sánh các dãy ký tự có chiều dài khác nhau, hay các dãy ký tự có xu hướng không chỉ bị thay thế không thôi, mà còn bị ảnh hưởng bởi dữ liệu bị lồng thêm vào hoặc bị xóa đi, phương pháp đo lường phức tạp hơn như khoảng cách Levenshtein (*Levenshtein distance*) là một phương pháp có tác dụng và thích hợp hơn.

Trọng số Hamming

Trọng số Hamming (Hamming weight) của một dãy ký tự là số phần tử trong dãy ký tự có giá trị khác không (0): đối với một dãy ký tự nhị phân (binary string), nó chỉ là số các ký tự có giá trị một (1), lấy ví dụ trọng số Hamming của dãy ký tự 11101 là 4.

Định nghĩa: Cho $V(v_1,v_2,v_3,..,v_n)$ là một vec tơ *n* thành phần nhị phân. Trọng số Hamming của *V*, ký hiệu là *W(V)* được định nghĩa là số thành phần khác 0 của *V*.

Ví dụ: Véc tơ $V(v)$ = 01011 thì $W(v)$ = 3

Khoảng cách Hamming

Trong lý thuyết thông tin, khoảng cách Hamming (Hamming distance) giữa hai dãy ký tự có chiều dài bằng nhau là số các ký hiệu ở vị trí tương đương có giá trị khác nhau. Nói một cách khác, khoảng cách Hamming đo số lượng thay thế cần phải có để đổi giá trị của một dãy ký tự sang một dãy ký tự khác hay là số lượng lỗi xảy ra khi biến đổi một dãy ký tự sang một dãy ký tự khác.

Ví dụ: Khoảng cách Hamming giữa 1011101 và 1001001 là 2.
Khoảng cách Hamming giữa 2143896 và 2233796 là 3.
Khoảng cách Hamming giữa "toned" và "roses" là 3.

Định nghĩa: Khoảng cách Hamming giữa hai véc tơ $U(u_1,u_2,u_3,..,u_n)$, $V(v_1,v_2,v_3,..,v_n)$ là số các toạ độ khác nhau giữa chúng. Ký hiệu là $D(U,V)$.

Ví dụ: Tính khoảng cách hai véc tơ 11001 và 10110 là 4

Từ định nghĩa khoảng cách Hamming giữa hai vec tơ ta có kết quả sau: $D(U,V)= W(U+V)$ trong đó $U+V$ là một vec tơ có được từ phép cộng modul 2 giữa hai vec tơ U và vec tơ V. U = 11001, V = 10110: $U + V$ = 01111. Như vậy khoảng cách của U và V là $W(U + V)$ = 4.

Cơ chế phát hiện sai bằng khoảng cách Hamming

Ta thấy khoảng cách giữa hai từ mã sẽ thay đổi 1 đơn vị nếu thay đổi 1 ký hiệu nào đó trong một từ mã. Khoảng cách bằng 0 nếu 2 từ mã trùng nhau. Vậy nếu 2 mã hiệu có khoảng cách D = 1 thì không phát hiện được sai 1 ký hiệu, muốn phát hiện sai 1 ký hiệu giữa hai từ mã phải có khoảng cách mã D tối thiểu bằng 2.

Vậy muốn phát hiện từ mã sai t ký hiệu thì khoảng cách mã phải thoả mãn điều kiện $D \geq t+1$.

Ví dụ1: Từ mã 1110 dưới tác dụng của từ mã sai 0001 sẽ thành 1111, như vậy nếu ta chọn a_1 = 1110, a_2 = 1111 (D =1) thì khi a_1 sai một ký hiệu thành '1111' khi nhận ta không phát hiện ra sai được.

Ví dụ 2: Giả sử có bộ mã: a_1 = 0011; a_2 = 0101; a_3 = 1100; a_4 = 1111

Bộ mã này có $D = 2$ khi một từ mã bất kỳ sai đi 1 ký hiệu ta nhận ra ngay. Giả sử từ mã a_1 sai 1 ký hiệu trở thành 0111, 1011, 0001, 0010 ta nhận thấy các tổ hợp này là tổ hợp không dùng vậy đã có hiện tượng sai.

Cơ chế sửa sai bằng khảng cách Haming

Bây giờ hãy xét điều kiện mã không những chỉ phát hiện sai mà còn sửa chữa được sai nhầm. Điều kiện này được thoả mãn nếu sự sai nhầm làm chuyển đổi từ mã ban đầu thành những tổ hợp mã gần từ mã đó hơn là bất cứ một từ mã nào khác.

Ta thấy khi từ mã sai đi t ký hiệu thì khoảng cách của nó so với một từ mã bất kỳ khác cũng lệch đi t đơn vị. Vậy một từ mã sai lệch đi so với từ mã gốc của nó khoảng cách nhỏ hơn D/2 thì ta sẽ quy về từ mã gốc của nó được. Do đó từ mã sai t ký hiệu, ta muốn xác định được tổ hợp sai thuộc từ mã nào thì khoảng cách mã phải thoả mãn điều kiện: $D \geq 2t + 1$.

Ví dụ: Cho bộ mã: $a_1 =$ 00000, $a_2 =$ 01101, $a_3 =$ 10110, $a_4 =$ 11011

Bộ mã này có $D = 3$ nên có thể sửa sai cho 1 ký hiệu. Giả sử từ mã a_2 bị sai 1 ký hiệu thành $a_2' =$ 01111. Ta tính khoảng cách mã của a_2' với các từ trong bộ mã trên có: $D(a_1,a_2') = 4$, $D(a_2',a_2) = 1$, $D(a_2',a_3) = 3$, $D(a_2',a_4) = 2$. Như vậy a_2' gần a_2 nhất ta sửa a_2' thành $a_2 =$ 01101. Rõ ràng việc sửa này là chính xác.

4.6.5 Một số biện pháp xây dựng bộ mã phát hiện sai và sửa sai

- ***Dùng Parity***

Khi xây dựng bảng tin để phát đi, ta thêm vào một bit vào cuối nội dung thông tin để tạo ra tổng số các bit mang trị 1 là chẵn (Parity chẵn) hay lẻ (Parity lẻ). Khi nhận tin ta kiểm tra tổng các bit 1 để xem có bị thay đổi không (chẵn-lẻ) nếu dùng Parity chẵn mà tổng số bit 1 lẻ thì ta biết nhận tin sai. Muốn khôi phục tin ta tách các bit parity ra khỏi tin.

Ví dụ: Ta có parity chẵn các bit parity thêm vào như sau:

Thông tin					Parity
0	1	1	0	1	1
1	1	1	0	1	**0**
0	1	1	0	0	**0**

- ***Mã khối***

Phương pháp này ta làm giống như dùng Parity nhưng mức độ cao hơn. Ta tạo tin thành từng khối và trên các dòng, cột ta đều thêm các bit parity theo cách trên. Như vậy ta kiểm tra hai lần theo dòng và theo cột nên khả năng phát hiện cao hơn nhiều, ngoài ra ta còn có khả năng sửa sai nếu xác định được toạ độ sai chính xác.

Ví dụ:

1	1	0	0	1	**1**
0	0	1	0	1	**0**
1	1	0	1	0	**1**
0	0	1	1	0	**0**
1	0	1	0	1	**1**
1	**0**	**1**	**0**	**1**	

- ***Mã đối xứng (Mã thuận nghịch)***

Giả sử ta có bộ mã với các từ mã $a_1, a_2, ..., a_n$. Xuất phát từ bộ mã trên ta xây dựng bộ mã mới như sau:

$b_1 = a_1 a_1^R; b_2 = a_2 a_2^R; b_3 = a_3 a_3^R; ..., b_n = a_n a_n^R$ trong đó a_k^R là xâu nghịch đảo của xâu $a_k (k = 1, 2, ..., n)$. Do tính chất đối xứng nên bộ mã đảm bảo việc phát hiện sai và sửa sai tương đối dễ dàng và chính xác.

Ví dụ:

Bộ mã gốc: $a_1 = 0100; a_2 = 100; a_3 = 1101; a_4 = 001$

Bộ mã đối xứng:

$b_1 = 01000010; b_2 = 100001; b_3 = 11011011; b_4 = 001100$

- ***Mã tỉ lệ***

Kí hiệu p là số bit có trị (0), q là số bit có trị (1) trong từ mã, ta đưa ra quy tắc xây dựng một bộ mã thỏa mãn tính chất $\frac{p}{q} = r$ trong đó r là một số cho trước được gọi là hệ số tỉ lệ, tính chất trên cần đúng với mọi từ mã. Với tính chất tỉ lệ, việc phát hiện sai và sửa sai cũng tương đối dễ dàng.

Ví dụ: Bộ mã với hệ số tỉ lệ $r = 2$

$$a_1 = 001; a_2 = 010; a_3 = 100; a_4 = 011000; a_5 = 110000$$

4.7 Mã tuyến tính

Phương pháp biểu diễn mã tuyến tính sử dụng các phép toán đại số tuyến tính nên cách biểu diễn mã là thuận tiện

4.7.1 Phương pháp xây dựng mã tuyến tính

Người ta sử dụng các phép toán của đại số tuyến tính để biểu diễn mã. Giả sử bộ mã V gồm các từ mã có độ dài n, mỗi từ mã được gọi là một vectơ mã gồm n thành phần. Cơ số mã là m (trong trường hợp nhị phân thì $m = 2$).

- ***Biểu diễn bằng ma trận sinh***

Xây dựng ma trận G gồm k hàng, mỗi hàng là một từ mã thuộc V được gọi là vectơ nền của mã V, khi đó G được gọi là ma trận sinh của mã V khi và chỉ khi bất kỳ một từ mã nào thuộc V cũng là một tổ hợp tuyến tính của các hàng của ma trận V.

V được gọi là không gian hàng của ma trận sinh G, nếu số chiều của không gian vectơ V là k thì ma trận G sẽ có k hàng.

Như vậy các tổ hợp tuyến tính khác nhau tương ứng với những vectơ mã khác nhau, Nếu V có k vectơ nền thì hệ gồm k vectơ đó sẽ là hệ độc lập tuyến tính.

Giả sử $G = \begin{bmatrix} a_1 \\ a_2 \\ .. \\ a_k \end{bmatrix}$ thì khi đó một vectơ mã $v = \alpha_1 a_1 + \alpha_2 a_1 + ... + \alpha_k a_k$. Trong đó α_i là các ký hiệu của mã. Mã $V(n,k)$ sẽ có m^k từ mã.

Ví dụ : Ta có ma trận sinh sau : $a_1 = 01101$; $a_2 = 10110$

$$G = \begin{bmatrix} 01101 \\ 10110 \end{bmatrix}$$

Mã $V(5,2)$ có $n = 5$ và $k = 2$, các từ mã như sau :

$$v = \alpha_1 a_1 + \alpha_2 a_2$$

$$V = \{00000;\ 01101;\ 10110;\ 11011\}$$

Ví dụ: Xác định mã $V(5,3)$ từ ma trận sinh G :

$$G = \begin{bmatrix} 10011 \\ 01010 \\ 00101 \end{bmatrix}$$

Ta thấy số tổ hợp là 2^5 sẽ có 32 tổ hợp nhưng ta chỉ dùng 8 tổ hợp (8 vec tơ mã) còn lại là những tổ hợp không dùng, mã này cũng cho phép ta phát hiện và sửa sai được. Cách biểu diễn này rất gọn và thuận tiện, nhất là khi các số n và k lớn trong khi nếu ta biểu diễn bằng bảng mã thì rất cồng kềnh và có thể không thực hiện được. Ngoài ra, ma trận sinh còn được gọi là ma trận sinh mã kiểm tra chẵn lẻ.

- ***Biểu diễn bằng ma trận thử***

Xây dựng một ma trận H gồm *n-k* hàng độc lập tuyến tính (trong đó k là số hàng của ma trận G), H là ma trận mà bất kỳ một vec tơ mã $v \in V$ thì $vH^T = 0$ (0 là vec tơ không, H^T là chuyển vị của H). Không gian vec tơ V' tạo bởi ma trận H được gọi là không gian không của không gian hàng V (hay V' là không gian hàng của ma trận H, không gian không của G). H được gọi là ma trận thử của mã V.

Mỗi vec tơ mã của V đều trực giao với mỗi vectơ thuộc $V'(n, n-k)$.

Ví dụ :

$$G = \begin{bmatrix} 10011 \\ 01010 \\ 00101 \end{bmatrix} \quad \text{có ma trận thử} \quad H = \begin{bmatrix} 11010 \\ 10101 \end{bmatrix}$$

• ***Mã hệ thống***

Xét bộ mã nhị phân. Nếu bằng một phép biến đổi ta chuyển ma trận G về dạng sau :

$$G' = \begin{bmatrix} 10...0\,p_{11}p_{12}...p_{1,n-k} \\ 01...0\,p_{2,1}p_{2,2}..p_{2,n-k} \\ 00...1\,p_{k,1}p_{k,2}..p_{k,n-k} \end{bmatrix}$$

Với mỗi bộ $\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)$ ta có một vectơ mã $v = \alpha G'$

$v = (\alpha_1, \alpha_2, ..., \alpha_k, C_1, ..., C_{n-k})$ trong đó $C_j = \sum_{i=1}^{k} \alpha_i p_{i,j}$

k thành phần đầu của v được gọi là những ký hiệu mang tin, $n-k$ thành phần sau được gọi là các ký hiệu dư hay ký hiệu thử (ký hiệu là r).

Ta thấy $n-k$ thành phần sau là tổ hợp của các thành phần đầu, vì vậy quá trình mã hoá đơn giản. Bộ mã này được gọi là mã hệ thống.

4.7.2 Nguyên lý giải mã

Trong nội dung trên, việc giải mã thực hiện đối chiếu vec tơ nhận được trong các lớp kề của bảng giải mã, tuy nhiên cách làm này không hiệu quả khi n, k lớn. Để khắc phục hạn chế ta thực hiện theo thuật toán dựa trên cơ sở tính Syndrom. Cho V là mã tuyến tính (n,k) bao gồm các từ mã là các vec tơ : h_1 (vec tơ không) và những vec tơ mã còn lại $h_2, h_3, ..., h_{2^k}$. Các vec tơ sai $e_1, e_2, ..., e_p$. Để thuận tiện trong quá trình lập thuật toán giải mã ta định nghĩa khái niệm lớp kề như sau :

Định nghĩa 1: V_j là lớp kề thứ j nếu $V_j = \{v : v = h_i + e_j, i = 1, 2, ..., 2^k\}$, e_j và v được gọi là vec tơ tạo lớp kề.

Định lý 1: Nếu bảng giải mã theo cách xếp lớp kề, khi nhận được vec tơ v sẽ giải mã đúng thành vec tơ u khi và chỉ khi vec tơ sai $v+u$ là vec tơ tạo lớp kề.

Định nghĩa 2: Syndrom của vec tơ v gồm r thành phần được xác định $S = vH^T$.

Khi v là từ mã đúng thì Syndrom sẽ bằng không, ngược lại Syndrom khác 0.

Định lý 2: Hai vec tơ v_1, v_2 cùng nằm trong một lớp kề khi và chỉ khi Syndrom của chúng bằng nhau.

Trên cơ sở các Định lý trên, ta có thuật toán giải mã :

Bước 1: Thiết lập bảng giải mã bằng cách xếp các lớp kề với những Syndrom tương ứng cho từng lớp.

Bước 2: Khi nhận được một dãy v' dài n thì tính Syndrom của v' và tìm lớp kề V_j ứng với Syndrom đã tính.

Bước 3: Từ mã gốc $v = v' + e_j$. e_j là vec tơ tạo lớp kề V_j.

Ví dụ: Cho ma trận sinh *G*

$$G = \begin{bmatrix} 10011 \\ 01010 \\ 00101 \end{bmatrix} \quad \text{có ma trận thử} \quad H = \begin{bmatrix} 11010 \\ 10101 \end{bmatrix}$$

Bước 1 : Xếp lớp kề :

v / e	00000	00101	01010	01111	10011	10110	11001	11100
V_1	00001	00100	01011	01110	10010	10111	11000	11101
V_2	00010	00111	01000	01101	10001	10100	11011	11110
V_3	10000	10101	11010	11111	00011	00110	01001	01100

Tính Syndrom của các lớp 1, 2, 3 lần lượt bằng : 01, 10, 11.

Các vec tơ tạo lớp kề : 00001, 00010, 10000 là những vec tơ sai. Mã này không sửa được hết tất cả các từ mã sai một bit, vì khoảng cách mã tối thiểu $D = 2$. Muốn sửa được thì $D = 3$.

Bước 2: Giả thiết nhận được $v' = 10111$, Syndrom $S = v'H^T = 01$ ứng với vec tơ tạo lớp kề 00001.

Bước 3: Từ mã gốc: $v = 10111 + 00001 = 10110$.

4.7.3 Một số giới hạn của mã tuyến tính

Trong một mã tuyến tính, khả năng sửa sai của mã là một tiêu chuẩn hàng đầu. Khả năng này được biểu thị bằng số ký hiệu sai tối đa có thể sửa được trong một tổ hợp mã và có liên quan đến trọng lượng tối thiểu của mã (khoảng cách mã). Cho nên khi xây dựng mã sửa sai tối ưu, cần phải xác định trước một số giới hạn của mã tuyến tính của mã như giới hạn trên về trọng lượng tối thiểu, giới hạn dưới về số ký hiệu thử đối với mã tuyến tính. Dưới đây là một số giới hạn đối với mã nhị phân.

- ***Giới hạn trên của khoảng cách mã tối thiểu***

Tổng trọng số của mã tuyến tính *(n, k)* như sau :

Mỗi từ mã có n ký hiệu, bộ mã có 2^k từ mã. Như vậy có tổng số ký hiệu là $n.2^k$. Trong đó có 1 ký hiệu khác 0. Giả thiết $p(0) = p(1) = \frac{1}{2}$. Tổng trọng số là: $n.2^k.\frac{1}{2} = n.2^{k-1}$.

Ta thấy khoảng cách mã của từ mã 0 đến từ mã có trọng số thấp nhất chính là trọng số của từ mã đó giả thiết đó là trọng số tối thiểu D. Rõ ràng trọng số tối thiểu thì phải nhỏ hơn trọng số trung bình. Từ đó ta có thể rút ra điều kiện cho khoảng cách mã D phải thoả mãn điều kiện: $D \le \frac{n2^{k-1}}{2^k - 1}$

Như vậy D phải nhỏ hơn hoặc lớn nhất là bằng tỷ số trên, đây chính là giới hạn trên của khoảng cách mã tối thiểu.

• ***Giới hạn Plotkin (về số ký hiệu thử)***

Điều kiện đã xét trên nếu r nhỏ, những ký hiệu mang tin sẽ có trị làm cho D vượt quá giới hạn đã xét trên. Bằng tính toán ta có giới hạn về số ký hiệu thử r sau gọi là giới hạn Plotkin: $r \geq 2D$ trong đó $r = n-k, k$ số ký hiệu mang tin, n là độ dài của từ mã.

• ***Giới hạn Haming***

Ta thấy bộ mã có cơ số 2, từ mã có độ dài *n* nếu có *t* ký hiệu sai thì số phương án sai là $\sum_{i=1}^{t} C_n^i$. Cả bộ mã sẽ có 2^k từ mã nên có số phương án sai là : $2^k \sum_{i=1}^{t} C_n^i$. Để phát hiện ra sai thì số tổ hợp cấm phải không ít hơn số các phương án sai ta có điều kiện sau:

$$N_0 - N \geq 2^k \sum_{i=1}^{t} C_n^i \Leftrightarrow \quad 2^{n-k} \geq 1 + \sum_{i=1}^{t} C_n^i \Leftrightarrow \quad 2^{n-k} \geq \sum_{i=0}^{t} C_n^i \Leftrightarrow$$

$$n - k \geq \log_2 \sum_{i=0}^{t} C_n^i$$

Giới hạn trên được gọi là giới hạn Haming.

Định lý: *Một mã bất kỳ dài n với trọng số tối thiểu $D = 2t+1$ hoặc lớn hơn phải có số ký hiệu thử r ít nhất là $\log_2 \sum_{i=0}^{t} C_n^i$.*

4.8 Mã vòng

4.8.1 Khái niệm:

Mã vòng là loại mã mà một từ mã có thể được biểu diễn dưới dạng một đa thức: $a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_2x^2 + a_1x^1 + a_0$ và có các đặc điểm sau:

• Nếu $a_0, a_1, a_2, ..., a_{n-1}$ là một từ mã thì tổ hợp được dịch chuyển một bậc $a_{n-1}, a_0, a_1, a_2, ..., a_{n-2}$ cũng sẽ là một tổ hợp mã.

• Những từ mã đều chia chẵn cho một đa thức $P(x)$ gọi là đa thức sinh (Đa thức tạo mã, từ đa thức này có thể tạo ra các từ mã của mã vòng xuất phát

từ mã đơn giản). Nếu bậc của mã đơn giản $A(x)$ là k, bậc của mã vòng $F(x)$ là n thì bậc của đa thức sinh $P(x)$ là $r=n-k$.

4.8.2 Nguyên lý lập mã

Nếu đa thức sinh được biểu diễn dưới dạng tổ hợp: $a_0, a_1, ..., a_r$, thì mã vòng có thể được biểu diễn bằng một ma trận sinh có dạng:

$$G = \left[\begin{array}{l} a_0\ a_1 \ldots a_r\ 0\ 0 \ldots\ldots 0 \\ 0\ a_0\ a_1 \ldots a_r\ 0 \ldots\ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\ 0 \ldots\ldots\ldots a_0\ a_1 .. \ 0 \\ \underbrace{0\ 0 \ldots\ldots\ldots 0}_{k-1}\ \underbrace{a_0 \ldots a_r}_{r+1} \end{array}\right] \Big\} k$$

Bằng những phép biến đổi tuyến tính ma trận G có thể được biến đổi về dạng chuẩn tắc sau:

$$G = \left[\begin{array}{l} p_{11}\ p_{12} \ldots\ldots p_{1r}\ 1\ 0 \ldots\ldots 0 \\ p_{21}\ p_{22} \ldots p_{2r}\ 0\ 1 \ldots\ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \underbrace{p_{k1}\ p_{k2} \ldots p_{kr}}_{r}\ \underbrace{0 \ldots\ldots\ldots 1}_{k} \end{array}\right]$$

Cách biểu diễn của ma trận sinh dưới dạng chuẩn tắc cho thấy: một từ mã gồm có hai phần, một phần có r ký hiệu thử và phần còn lại có k ký hiệu mang tin. Để xác định đa thức sinh cần dựa trên tính chất chia chẵn nhị thức $1+x^n$.

Từ những tính chất trên có thể trực tiếp xây dựng ma trận sinh dạng chuẩn tắc bằng cách chia x^n cho $P(x)$, các số thừa trong phép chia sẽ thành một ma trận $C_{n-k,k}$, hợp với ma trận $\tilde{I}_k$.

$$\tilde{I}_k = \begin{bmatrix} 0\,0....0\,1 \\ 0\,0....1\,0 \\ ..... \\ 0\,1....0\,0 \\ 1\,0....0\,0 \end{bmatrix}$$ tạo thành ma trận sinh $G^*_{n,k} = \left[\widetilde{I}_k, C_{n-k,k}\right]$

Ví dụ : Xây dựng một mã vòng $(7,4)$.

Trước hết xác định đa thức sinh $P(x)$ lấy trong những thừa số chung của nhị thức:

$$x^7 + 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1).$$

$r = n - k = 3 \Rightarrow P(x) = x^3 + x^2 + 1$, viết dưới dạng tổ hợp nhị phân là: 1101.

Xác định ma trận các số thừa $C_{3,4}$ của phép chia x^7 cho $P(x)$. Sau khi thực hiện phép chia nhị phân ta được ma trận:

$$C_{3,4} = \begin{bmatrix} 1\ 0\ 1 \\ 1\ 1\ 1 \\ 0\ 1\ 1 \\ 1\ 1\ 0 \end{bmatrix} \Rightarrow G^*_{7,4} = \begin{bmatrix} 0\,0\,0\,1\,1\ 0\,1 \\ 0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,0\,0\,0\,1\,1 \\ 1\,0\,0\,0\,1\,1\,0 \end{bmatrix}$$

Để có thể thực hiện một cách thuận tiện các thiết bị tạo mã và giải mã, cần có thuật toán lập và giải mã. Các thuật toán này đều dựa trên tính chất cơ bản là các từ mã chia chẵn cho đa thức sinh. Từ mã tìm được: $F(x) = P(x).Q(x) = x^{n-k}A(x) + R(x)$ trong đó $Q(x) = \dfrac{x^{n-k}A(x)}{P(x)} + \dfrac{R(x)}{P(x)}$.

Ví dụ : Tổ hợp mã đơn giản $A(x) = x^2 + 1, x^{n-k} = x^2$, $x^{n-k}.A(x) = x^5 + x^3$. Đem chia $x^5 + x^3$ cho $P(x) = x^3 + x^2 + 1$ ta được: $\dfrac{x^5 + x^3}{x^3 + x^2 + 1} = x^2 + x + \dfrac{x^2 + x}{x^3 + x^2 + 1}$.

Vì vậy tổ hợp mã sẽ là: $x^5 + x^3 + x^2 + x$, viết dưới dạng tổ hợp mã nhị phân là 101110.

4.8.3 Nguyên lý giải mã

Đem chia từ mã nhận được cho đa thức $P(x)$, nếu phép chia chẵn chứng tỏ từ mã thu được là đúng, trong trường hợp phép chia còn số thừa thì chứng tỏ từ mã thu được có ký hiệu sai. Số thừa cho phép xác định vị trí của ký hiệu sai. Vec tơ gốc $u = v + e$. Trong đó *v* là vec tơ thu được, *e* là vec tơ sai.

Phương pháp thực hiện sửa sai có thể dựa trên ma trận phát hiện và sửa sai

$$M_{2n-k,n} = \begin{bmatrix} \tilde{I}_{n-k} \\ \tilde{I}_n \\ C_{n-k,k} \end{bmatrix}$$

Ví dụ: Đối với mã vòng $(7,4)$ ta có các ma trận sau:

$$C_{3,4} = \begin{bmatrix} 1\ 0\ 1 \\ 1\ 1\ 1 \\ 0\ 1\ 1 \\ 1\ 1\ 0 \end{bmatrix} \qquad \tilde{I}_3 = \begin{bmatrix} 0\ 0\ 1 \\ 0\ 1\ 0 \\ 1\ 0\ 0 \end{bmatrix} \qquad \tilde{I}_7 = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,0\,0\,0\,0 \end{bmatrix}$$

Các hàng của ma trận $\tilde{I}_7$ đại biểu cho các vec tơ sai, vị trí của số đơn vị cho biết vị trí của ký hiệu sai trong tổ hợp mã tuyến tính từ trái sang phải theo bậc từ cao giảm xuống thấp: $x^6, x^5, x^4, x^3, x^2, x^1, x^0$.

Ví dụ: hàng thứ ba của ma trận số đơn vị ở vị trí thứ 5 tương ứng với ký hiệu sai x^2. Ma trận phát hiện sai và sửa sai của mã $(7,4)$ là:

$$M_{10,7} = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,1\;0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\;0\,1\,0 \\ 0\,0\,0\,0\,1\,0\,0\;1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\;1\,0\,1 \\ 0\,0\,1\,0\,0\,0\,0\;1\,1\,1 \\ 0\,1\,0\,0\,0\,0\,0\;0\,1\,1 \\ 1\,0\,0\,0\,0\,0\,0\;1\,1\,0 \end{bmatrix}$$

Giả sử thu được từ mã: 0111100, trong khi đó ở đầu phát đã gửi đi từ mã: 0110100. Như vậy trong quá trình truyền tin đã gây sai ký hiệu thứ 4 kể từ trái. Quá trình phát hiện sai và sửa sai được thực hiện như sau:

Đa thức sinh được chọn như trước $P(x) = x^3 + x^2 + 1$. Tiến hành kiểm tra tổ hợp mã thu được là đúng hay sai bằng cách đem chia từ mã đó cho đa thức sinh $P(x) = x^3 + x^2 + 1$. Sau khi thực hiện phép chia được số thừa là: 101, đem đối chiếu trong ma trận $M_{10,3}$ thấy số thừa thuộc hàng thứ tư tương ứng với vec tơ sai: 0001000, điều này cho thấy bit thứ tư của từ mã đã sai. Vec tơ gốc = 0111100+0001000 =0110100.

Để thực hiện các sơ đồ tạo mã và giải mã vòng, sử dụng các mạch nhân, chia đa thức với đa thức. Các mạch này được xây dựng trên cơ sở các phần tử cơ bản của mạch như sau:

- $\rightarrow \oplus \leftarrow$: Cộng modun 2
- $\rightarrow a_i \rightarrow$: Nhân với một hệ số bằng a_i
- $\rightarrow \square \leftarrow$: Mạch ghi chuyển

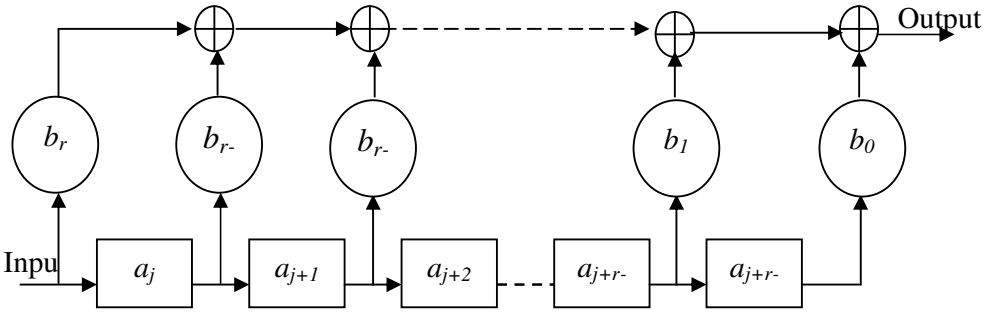
Sơ đồ nhân hoặc chia một đa thức cho một đa thức là những dạng tổ hợp khác nhau của các phần tử cơ bản trên được mô tả như sau:

Sơ đồ nhân một đa thức với một đa thức: $A(x).P(x)$

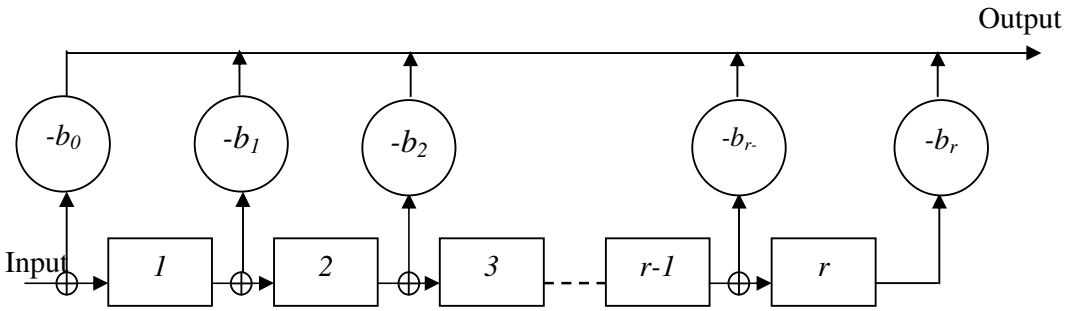
$$A(x) = a_0 + a_1 x + ... + a_i x^i$$

$$P(x) = b_0 + b_1 x + ... + b_r x^r$$

$$A(x)P(x) = a_0 b_0 + (a_0 b_1 + a_1 b_0)x + (a_0 b_2 + a_1 b_1 + a_2 b_0)x^2 + ... + (a_{i-1} b_r + a_i b_{r-1})x^{i+r-1} + (a_i b_r)x^{1+r}$$



Sơ đồ chia một đa thức cho một đa thức:



CHƯƠNG 5. GIỚI THIỆU VỀ HỆ MẬT MÃ

5.1 Khái niệm hệ mật mã

Đối tượng cơ bản của mật mã là tạo ra khả năng liên lạc trên một kênh cho hai người sử dụng (Giả sử là người A và người B) sao cho đối phương không thể hiểu được thông tin được truyền đi. Kênh này có thể là một đường dây điện thoại hoặc một mạng máy tính. Thông tin mà B muốn gửi cho A (bản rõ) có thể là một văn bản tiếng Anh, các dữ liệu bằng số hoặc bất cứ tài liệu nào có cấu trúc tuỳ ý. Khi đó B sẽ mã hoá bản rõ bằng một khoá đã được xác định trước và gửi bản mã kết quả trên kênh. Đối phương có bản mã thu được trên kênh song không thể xác định nội dung của bản rõ, nhưng A (người đã biết khoá mã) có thể giải mã và thu được bản rõ.

Ta sẽ mô tả mô hình toán học tổng quát như sau:

Một hệ mật là một bộ 5 (P, C, K, E, D) thoả mãn các điều kiện sau:

+ P (Plaintext): Là một tập hữu hạn các bản rõ có thể.

+ C (Ciphertext): Là một tập hữu hạn các bản mã có thể.

+ K (Key): Là tập hữu hạn các khoá có thể.

+ E (Encryption): Là tập các hàm mã hoá.

+ D (Decryption): Là tập các hàm giải mã.

Đối với mỗi $k \in K$ có một quy tắc mã e_k: $P \rightarrow C$ và một quy tắc giải mã tương ứng $d_k \in D$.

Mỗi e_k: $P \rightarrow C$ và d_k: $C \rightarrow P$ là những hàm mà $d_k (e_k (x)) = x$ với mọi bản rõ $x \in P$.

5.2 Phân loại các hệ thống mật mã

- Theo cơ chế mã hoá và giải mã:

Hệ mã cổ điển (hệ mã đối xứng): dùng 1 khoá để mã hoá và giải mã.

Hệ mã hiện đại (hệ mã bất đối xứng): dùng một khoá để mã hoá và 1 khoá để giải mã. Khoá mã hoá có thể công khai, khoá giải mã phải giữ bí mật.

• Theo cách mã hoá:

Mã khối: mã hoá sử dụng các thuật toán khối, dữ liệu được chia thành khối trước khi mã với kích thước tuỳ ý nhưng phải cố định.

Mã dòng: là các thuật toán mã hoá và giải mã thực hiện theo từng bit tại mỗi thời điểm.

5.3 Hệ mã cổ điển (Symmetric-key encryption)

5.3.1 Khái niệm

Hệ mã cổ điển là loại mã được thực hiện thông qua hàm f có tính thuận nghịch, sử dụng f để mã hoá, biết f có thể suy ra hàm giải mã f^{-1}. Đây là hệ mã dùng cùng một khoá để mã hoá và giải mã, khoá phải được giữ bí mật.

5.3.2 Một số hệ mã cổ điển

• ***Mã hoán vị (MHV)***

Ý tưởng của MHV là giữ các ký tự của bản rõ không thay đổi nhưng sẽ thay đổi vị trí của chúng bằng cách sắp xếp lại các ký tự này. MHV (còn được gọi là mã chuyển vị) đã được dùng từ hàng trăm năm nay. Giả sử m là một số nguyên dương xác định nào đó, kí hiệu $P = C = (Z_{26})^m$ và cho tất cả các hoán vị của {1, . . ., m}. Đối với một khoá π (tức là một hoán vị) ta xác định:

$$\begin{cases} e_\pi(x_1, \ldots, x_m) = (x_{\pi(1)}, \ldots, x_{\pi(m)}), \\ d_\pi(x_1, \ldots, x_m) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)}) \end{cases}$$

trong đó π^{-1} là hoán vị ngược của π.

Giả sử ta có bản rõ là : khoacongnghethongtindaihocthainguyen.

Trước tiên ta nhóm bản rõ thành các nhóm 6 ký tự :

khoaco | ngheth | ongtin | daihoc | thaing | uyenzz

Mỗi nhóm 6 chữ cái được sắp xếp lại theo phép hoán vị π, ta có:

AOKOHC| EHNHGT | HIDCAO | IATGHN | NEUZYZ

Như vậy bản mã là :

AOKOHCEHNHGTHIDCAOIATGHNNEUZYZ

Việc thực hiện giải mã đực thực hiện thông qua hoán vị đảo π^{-1}

- ***Mã dịch vòng (shift cipher)***

Trong phần này sẽ mô tả mã dịch (MD) dựa trên lý thuyết số học theo modulo, chúng ta sẽ xét một số định nghĩa cơ bản của số học liên quan đến vấn đề này.

Định nghĩa: Giả sử a và b là các số nguyên và m là một số nguyên dương, kí hiệu $a \equiv b \pmod m$ nếu m chia hết cho b-a. Mệnh đề $a \equiv b \pmod m$ được gọi là "a đồng dư với b theo modulo m".

Giả sử chia a và b cho m và ta thu được thương nguyên và phần dư, các phần dư nằm giữa 0 và m-1, nghĩa là $a = q_1m + r_1$ và $b = q_2m + r_2$ trong đó $0 \le r_1 \le m-1$ và $0 \le r_2 \le m-1$. Khi đó có thể dễ dàng thấy rằng $a \equiv b \pmod m$ khi và chỉ khi $r_1 = r_2$. Ta sẽ dùng ký hiệu a mod m để xác định phần dư khi a được chia cho m. Như vậy $a \equiv b \pmod m$ khi và chỉ khi a mod m = b mod m. Nếu thay a bằng a mod m thì ta nói rằng a được rút gọn theo modulo m.

Nhận xét: Nhiều ngôn ngữ lập trình của máy tính xác định a mod m là phần dư trong khoảng (-m+1),..., (m-1) có cùng dấu với a, ví dụ -18 mod 7 sẽ là -4, giá trị này khác với giá trị 3 là giá trị được xác định theo công thức trên. Tuy nhiên, để thuận tiện ta sẽ xác định a mod m luôn là một số không âm.

Bây giờ ta định nghĩa modulo m kí hiệu là Z_m là tập hợp {0,1,. . .,m-1} có trang bị hai phép toán cộng và nhân, việc cộng và nhân trong Z_m được thực hiện giống như cộng và nhân các số thực với các kết quả được rút gọn theo modulo m.

Ví dụ tính 11× 13 trong Z_{16} . Tương tự như với các số nguyên ta có 11 ×13 = 143. Để rút gọn 143 theo modulo 16, ta thực hiện phép chia bình thường: 143 = 8 × 16 + 15, bởi vậy 143 mod 16 = 15 trong Z_{16} .

Định nghĩa mã dịch vòng

Giả sử P = C = K = Z_{26} với 0 ≤ k ≤ 25, định nghĩa mã dịch vòng dạng toán học như sau:

$$\begin{cases} e_k(x) = x + k \pmod{26} \\ d_k(x) = y - k \pmod{26} \end{cases} \quad (\text{với } x,y \in Z_{26})$$

Nhận xét: Trong trường hợp K = 3, hệ mật thường được gọi là mã Caesar đã từng được Julius Caesar sử dụng.

Ta sẽ sử dụng MDV (với modulo 26) để mã hoá một văn bản tiếng Anh thông thường bằng cách thiết lập sự tương ứng giữa các kí tự và các thặng dư theo modulo 26 như sau: A ↔ 0,B ↔ 1, . . ., Z ↔ 25.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Ví dụ:Giả sử khoá cho MDV là K = 5 và bản rõ là: Khoacongnghethongtin. Trước tiên biến đổi bản rõ thành dãy các số nguyên nhờ dùng phép tương ứng trên:

10	7	14	0	2	14	13	6	13	6
7	4	19	7	14	13	6	19	8	13

sau đó cộng 5 vào mỗi giá trị rồi rút gọn tổng theo modulo 26

15	12	19	5	7	19	18	11	18	11
12	9	24	12	19	18	11	24	13	18

Cuối cùng biến đổi dãy số nguyên này thành các kí tự thu được bản mã sau: PMTFHTSLSLMJYMTSLYNS

Để giải mã bản mã này, ta sẽ biến đổi bản mã thành dãy các số nguyên rồi trừ đi giá trị cho 11 (rút gọn theo modulo 26) và cuối cùng biến đổi lại dãy này thành các ký tự.

Chú ý: Trong ví dụ trên, ta đã dùng các chữ in hoa cho bản mã, các chữ thường cho bản rõ để tiện phân biệt. Quy tắc này còn tiếp tục sử dụng sau này.

Nhận xét: MDV (theo modulo 26) là không an toàn vì nó có thể bị thám theo phương pháp vét cạn. Do chỉ có 26 khoá nên dễ dàng thử mọi khoá d_K có thể cho tới khi nhận được bản rõ có nghĩa.

- ***Mã thay thế (MTT)***

Cho P =C = Z_{26}, K chứa mọi hoán vị của 26 kí hiệu 0,1,...,25. Với mỗi phép hoán vị $\pi \in K$, kí hiệu

$$\begin{cases} e_\pi(x) = \pi(x) \\ d_\pi(y) = \pi^{-1}(y) \end{cases}$$

trong đó π^{-1} là hoán vị ngược của π.

Sau đây là một ví dụ về phép hoán vị ngẫu nhiên π tạo nên một hàm mã hoá:

a	b	c	d	e	f	g	h	i	j	k	l	M
X	N	Y	A	H	P	O	G	Z	Q	W	B	T

n	o	p	q	r	s	t	u	v	w	x	y	Z
S	F	L	R	C	V	M	U	E	K	J	D	I

Như vậy, e_π (a)=X ,e_π(b) = N,. . . . Hàm giải mã là phép hoán vị ngược. Điều này được thực hiện bằng cách viết hàng thứ hai lên trước rồi sắp xếp theo thứ tự chữ cái, ta nhận được:

A	B	C	D	E	F	G	H	I	J	K	L	M
d	l	r	y	v	o	h	e	z	x	w	p	T

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	g	f	j	q	n	m	u	s	k	a	c	I

Như vậy d_π (A) = d, d_π (B) = l . . .

Mỗi khoá của MTT là một phép hoán vị của 26 kí tự. Số các hoán vị này là 26!, đây là một số rất lớn vì vậy việc tìm khoá theo thuật toán vét cạn không thể thực hiện được. Tuy nhiên sau này sẽ thấy rằng MTT có thể dễ dàng bị thám bằng các thuật toán khác.

• ***Mã Affine***

MDV là một trường hợp đặc biệt của MTT chỉ gồm 26 trong số 26! các hoán vị có thể của 26 phần tử. Một trường hợp đặc biệt khác của MTT là mã Affine, ta giới hạn xét các hàm mã có dạng

$$e(x) = ax + b \bmod 26,\ a,b \in Z_{26}\ .$$

Các hàm này được gọi là các hàm Affine (chú ý rằng khi a = 1, ta có MDV). Để việc giải mã có thể thực hiện được, yêu cầu cần thiết là hàm Affine phải là đơn ánh. Nói cách khác, với bất kỳ $y \in Z_{26}$, phương trình

$$ax + b \equiv y \pmod{26}$$

có nghiệm x duy nhất. Điều này tương đương với:

$$ax \equiv y-b \pmod{26}.$$

Vì y thay đổi trên Z_{26} nên y-b cũng thay đổi trên Z_{26}. Vì vậy ta chỉ cần nghiên cứu phương trình đồng dư:

$$ax \equiv y \pmod{26}, \qquad (y \in Z_{26}).$$

Định lý: Đồng dư thức $ax \equiv b \bmod m$ chỉ có một nghiệm duy nhất $x \in Z_m$ với mọi $b \in Z_m$ khi và chỉ khi UCLN(a,m) = 1.

Vì $26=2\times13$ nên các giá trị $a \in Z_{26}$ thoả mãn UCLN(a,26)=1 là a=1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 và 25. Tham số b có thể là một phần tử bất kỳ trong Z_{26}. Như vậy, mã Affine có $12\times26 = 312$ khoá có thể. Bây giờ ta sẽ xét bài toán chung với modulo m.

Định nghĩa 1: Giả sử $a \geq 1$ và $m \geq 2$ là các số nguyên. UCLN(a,m) = 1 thì ta nói rằng a và m là nguyên tố cùng nhau. Số các số nguyên trong Z_m nguyên tố cùng nhau với m thường được ký hiệu là $\phi(m)$ (hàm này được gọi là hàm Euler).

Một kết quả quan trọng trong lý thuyết số cho ta giá trị của $\phi(m)$ theo các thừa số trong phép phân tích theo luỹ thừa các số nguyên tố của m. Một số nguyên p >1 là số nguyên tố nếu nó không có ước dương nào khác ngoài 1 và p. Mọi số nguyên m >1 có thể phân tích được thành tích của các luỹ thừa các số nguyên tố theo cách duy nhất. Ví dụ $60 = 2^3 \times 3 \times 5$ và $98 = 2 \times 7^2$.

Bây giờ ta sẽ xét xem các phép toán giải mã trong mật mã Affine với modulo m=26. Giả sử UCLN(a,26)=1. Để giải mã cần giải phương trình đồng

dư y ≡ax+b (mod 26) theo x. Từ trên ta thấy rằng phương trình này có một nghiệm duy nhất trong Z_{26}. Điều quan trọng là cần xác định một thuật toán để xác định nghiệm của phương trình.

Định nghĩa 2: Giả sử $a \in Z_m$. Phần tử nghịch đảo (theo phép nhân) của a là phần tử $a^{-1} \in Z_m$ sao cho $aa^{-1} \equiv a^{-1}a \equiv 1 \pmod m$.

Bằng các lý luận tương tự như trên, có thể chứng tỏ rằng a có nghịch đảo theo modulo m khi và chỉ khi UCLN(a,m) =1, và nếu nghịch đảo này tồn tại thì nó phải là duy nhất. Ta cũng thấy rằng, nếu $b = a^{-1}$ thì $a = b^{-1}$.

Sau đây chúng ta sẽ mô tả một thuật toán hữu hiệu để tính các nghịch đảo của Z_m với m tuỳ ý.

Xét phương trình đồng dư y ≡ ax+b (mod 26). Phương trình này tương đương với phương trình ax≡y-b (mod 26). Vì UCLN(a,26) =1 nên a có nghịch đảo theo modulo 26. Nhân cả hai vế của đồng dư thức với a^{-1} ta có

$$a^{-1}(ax) \equiv a^{-1}(y-b) \pmod{26} .$$

Sử dụng tính chất kết hợp của phép nhân modulo, ta có

$$a^{-1}(ax) \equiv (a^{-1}a)x \equiv 1x \equiv x.$$

Từ đó $x \equiv a^{-1}(y-b) \pmod{26}$ hay hàm giải mã là:

$$d(y) = a^{-1}(y-b) \bmod 26$$

- ***Mã Vigenère***

Trong cả hai hệ MDV và MTT, mỗi ký tự sẽ được ánh xạ vào một ký tự duy nhất. Vì lý do đó, các hệ mật còn được gọi hệ thay thế đơn biểu. Sau đây ta sẽ trình bày một hệ mật không phải là bộ chữ đơn, đó là hệ mã Vigenère. Mật mã này do Blaise de Vigenère đề xuất vào thế kỷ XVI.

Sử dụng phép tương ứng A ⇔ 0, B ⇔ 1, . . . , Z ⇔ 25 ở trên, ta có thể gắn cho mỗi khoá K với một chuỗi kí tự có độ dài m được gọi là từ khoá. Mật

mã Vigenère sẽ mã hoá đồng thời m kí tự: Mỗi phần tử của bản rõ tương đương với m ký tự.

Định nghĩa: Cho m là một số nguyên dương cố định nào đó, với khoá K = (k_1, k_2, . . . , k_m) kí hiệu

$$\begin{cases} e_k(x_1, x_2, \ldots, x_m) = (x_1+k_1, x_2+k_2, \ldots, x_m+k_m) \\ d_k(y_1, y_2, \ldots, y_m) = (y_1-k_1, y_2-k_2, \ldots, y_m-k_m) \end{cases}$$

trong đó tất cả các phép toán được thực hiện trong Z_{26}.

Ví dụ: Giả sử m =6 và từ khoá là CIPHER. Từ khoá này tương ứng với dãy số K = (2,8,15,4,17).

Giả sử bản rõ là xâu: *thiscryptosystemisnotsecure*

Ta sẽ biến đổi các phần tử của bản rõ thành các thặng dư theo modulo 26, viết chúng thành các nhóm 6 rồi cộng với từ khoá theo modulo 26 như sau:

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15
18	19	4	12	8	18	13	14	19	18	4	2
2	8	15	7	4	17	2	8	15	7	4	17
20	1	19	19	12	9	15	22	8	15	8	19

20	17	4
2	8	15
22	25	19

Do đó dãy kí tự của xâu bản mã sẽ là

V P X Z G I A X I V W P U B T T M J P W I Z I T W Z T

Để giải mã ta có thể dùng cùng từ khoá nhưng thay cho cộng, ta trừ cho nó theo modulo 26.

Nhận xét: Ta thấy rằng các từ khoá có thể với số độ dài m trong mật mã Vigenère là 26^m, với các giá trị m khá nhỏ, phương pháp tìm kiếm vét cạn cũng yêu cầu thời gian khá lớn. Trong hệ mật Vigenère có từ khoá độ dài m, mỗi ký tự có thể được ánh xạ vào trong m ký tự có thể có, một hệ mật như vậy được gọi là hệ mật thay thế đa biểu (polyalphabetic). Nói chung, việc thám mã hệ thay thế đa biểu sẽ khó khăn hơn so việc thám mã hệ đơn biểu.

• ***Mật mã Hill (Do Lester S.Hill đưa ra năm 1929).***

Giả sử m là một số nguyên dương, đặt $P = C = (Z_{26})^m$, ý tưởng ở đây là lấy m tổ hợp tuyến tính của m ký tự trong một phần tử của bản rõ để tạo ra m ký tự ở một phần tử của bản mã. Ví dụ nếu m = 2 ta có thể viết một phần tử của bản rõ là $x=(x_1,x_2)$ và một phần tử của bản mã là $y=(y_1,y_2)$. ở đây, y_1 và y_2 đều là các tổ hợp tuyến tính của x_1 và x_2. Chẳng hạn, có thể lấy

$$y_1 = 11x_1 + 3x_2$$
$$y_2 = 8x_1 + 7x_2$$

Hay biểu diễn dưới dạng ma trận

$$(y_1, y_2) = (x_1, x_2) \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$$

Tổng quát: ta có thể lấy một ma trận K kích thước m×m làm khoá, với $x=(x_1, x_2, \ldots, x_m) \in P$ và $K \in K$, ta tính $y = e_k(x) = (y_1, y_2, \ldots, y_m)$ theo công thức $y = xK$, hiển nhiên $x=yK^{-1}$.

Cho tới lúc này ta đã chỉ ra rằng có thể thực hiện phép giải mã khi và chỉ khi K là ma trận khả nghịch. Tính khả nghịch của một ma trận vuông phụ thuộc vào giá trị định thức của nó.

Nhận xét: Định thức của một ma trận vuông cấp m×m có thể được tính theo các phép toán sơ cấp. Trên Z_{26}, ta có ma trận K có nghịch đảo theo modulo 26

khi và chỉ khi UCLN(det K,26) = 1. Sau đây sẽ chứng minh ngắn gọn kết quả này.

Trước tiên, giả sử rằng UCLN(det K,26) = 1. Khi đó det K có nghịch đảo trong Z_{26} . Với $1 \le i \le m$, $1 \le j \le m$, định nghĩa K_{ij} là ma trận thu được từ ma trận K bằng cách loại bỏ hàng thứ i và cột thứ j, ma trận K^* có phần tử (i,j) của nó nhận giá trị bằng (-1)det $K_{ij,}$ (K^* được gọi là ma trận bù đại số của K). Khi đó có thể chứng tỏ rằng: $K^{-1} = (\det K)^{-1}K^*$, Như vậy K là khả nghịch.

Ngược lại K có nghịch đảo K^{-1} . Theo quy tắc nhân của định thức

$$1 = \det I = \det (KK^{-1}) = \det K \det K^{-1}$$

Như vậy det K có nghịch đảo trong Z_{26} .

• ***Các hệ mã dòng***

Trong các hệ mật nghiên cứu ở trên, các phần tử liên tiếp của bản rõ đều được mã hoá bằng cùng một khoá K tức là xâu bản mã y nhận được có dạng:

$$y = y_1y_2 \ldots = eK(x_1)\ eK(x_2)\ \ldots$$

Các hệ mật thuộc dạng này thường được gọi là các mã khối. Một quan điểm khác là mật mã dòng, mục đích chính là tạo ra một dòng khoá $z = z_1z_2 \ldots$ và dùng nó để mã hoá một xâu bản rõ $x = x_1x_2 \ldots$ theo quy tắc:

$$y = y_1y_2 \ldots = ez_1(x_1)\ ez_2(x_1) \ldots$$

Mã dòng hoạt động như sau: Giả sử $K \in K$ là khoá và $x = x_1x_2 \ldots$ là xâu bản rõ. Hàm f_i được dùng để tạo z_i (z_i là phần tử thứ i của dòng khoá) trong đó f_i là một hàm của khoá K và i-1 là ký tự đầu tiên của bản rõ:

$$z_i = f_i (K, x_1 , \ldots , x_{i-1})$$

Phần tử z_i của dòng khoá được dùng để mã x_i tạo ra $y_i = e_iz(x_i)$. Bởi vậy, để mã hoá xâu bản rõ $x_1\ x_2 \ldots$ ta phải tính liên tiếp: z_1, y_1, z_2 , y_2 ...

Việc giải mã xâu bản mã $y_1y_2 \ldots$ có thể được thực hiện bằng cách tính liên tiếp: z_1, x_1, z_2 , x_2 ...

Định nghĩa: Mật mã dòng là một bộ (P,C,K,L,F,E,D) thoả mãn các điều kiện sau:

1. P là một tập hữu hạn các bản rõ có thể.
2. C là tập hữu hạn các bản mã có thể.
3. K là tập hữu hạn các khoá có thể (không gian khoá)
4. L là tập hữu hạn các bộ chữ của dòng khoá.
5. F = (f_1 f_2...) là bộ tạo dòng khoá. Với i ≥ 1
6. $f_i : K \times P^{i-1} \rightarrow L$

Với mỗi z ∈L có một quy tắc mã $e_z \in E$ và một quy tắc giải mã tương ứng $d_z \in D$. $e_z : P \rightarrow C$ và $d_z: C \rightarrow P$ là các hàm thoả mãn $d_z(e_z(x))= x$ với mọi bản rõ $x \in P$.

Ta có thể coi mã khối là một trường hợp đặc biệt của mã dòng trong đó dùng khoá không đổi: $Z_i = K$ với mọi i ≥1.

Sau đây là một số dạng đặc biệt của mã dòng:

+ Mã dòng được gọi là đồng bộ nếu dòng khoá không phụ thuộc vào xâu bản rõ, tức là nếu dòng khoá được tạo ra chỉ là hàm của khoá K.

+ Một hệ mã dòng được gọi là tuần hoàn với chu kỳ d nếu $z_{i+d}= z_i$ với số nguyên i ≥ 1. Mã Vigenère với độ dài từ khoá m có thể coi là mã dòng tuần hoàn với chu kỳ m. Trong trường hợp này, khoá là K = (k_1, . . . k_m). Bản thân K sẽ tạo m phần tử đầu tiên của dòng khoá: $z_i = k_i$, $1 \le i \le m$, sau đó dòng khoá sẽ tự lặp lại. Nhận thấy rằng, trong mã dòng tương ứng với mật mã Vigenère, các hàm mã và giải mã được dùng giống như các hàm mã và giải mã được dùng trong MDV

$$e_z(x) = x+z \text{ và } d_z(y) = y-z$$

Các mã dòng thường được mô tả trong các bộ chữ nhị phân tức là $P = C = L = Z_2$. Trong trường hợp này, các phép toán mã và giải mã là phép cộng theo modulo 2.

$$e_z(x) = x + z \bmod 2 \quad \text{và} \quad d_z(x) = y + z \bmod 2.$$

Nếu kí hiệu "0" kí hiệu giá trị "sai" và "1" kí hiệu giá trị "đúng" thì phép cộng theo moulo 2 sẽ ứng với phép hoặc có loại trừ. Bởi vậy phép mã (và giải mã) dễ dàng thực hiện bằng việc thiết kế các vi mạch.

5.3.3 Hệ mã DES

Ý tưởng của hệ mật Des là tạo ra một thuật toán biến đổi dữ liệu thật phức tạp để đối phương không thể tìm ra mối liên quan của đoạn tin mã với bản rõ cũng như không thể thiết lập được mối quan hệ nào giữa đoạn tin được mã hóa và khóa.

Thuật toán tiến hành theo 3 giai đoạn:

1. Bản rõ cho trước x (64 bit), một xâu bit x_0 sẽ được xây dựng bằng cách hoán vị các bit của x theo phép hoán vị cố định ban đầu IP, khối dữ liệu chia thành 2 nửa (nửa trái và nửa phải). Ta viết: $x_0 = IP(X) = L_0R_0$, (L_0 gồm 32 bit đầu và R_0 là 32 bit cuối).

2. Tính toán 16 lần lặp theo một hàm xác định, xác định L_iR_i, $1 \le i \le 16$ theo quy tắc sau:

$L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1},K_i)$ trong đó kí hiệu $\oplus$ là phép hoặc loại trừ của hai xâu bit (cộng theo modulo 2), f là một hàm mà ta sẽ mô tả ở sau.

$K_1,K_2, \ldots ,K_{16}$ là các xâu bit độ dài 48 được tính như hàm của khoá K. (trên thực tế mỗi K_i là một phép chọn hoán vị bit trong K). $K_1, \ldots, K_{16}$ sẽ tạo thành bảng khoá.

3. Áp dụng phép hoán vị ngược IP^{-1} cho xâu bit $R_{16}L_{16}$, ta thu được bản mã $y = IP^{-1}(R_{16}L_{16})$. (Cần chú ý thứ tự đã đảo của L_{16} và R_{16}).

Như vậy để thực hiện mã hóa xâu bản rõ x thì ta phải xác định:

- Phép hoán vị IP. (cho sẵn)
- Tính được Li và Ri. (Tính hàm f và tính khóa K_i)
- Phép hoán vị đảo: IP^{-1} (cho sẵn)

Tính bảng khóa

Khóa K là một xâu bit độ dài 64, trong đó 56 bit là khoá và 8 bit để kiểm tra tính chẵn lẻ nhằm phát hiện sai. Các bit ở các vị trí 8,16, . . ., 64 được xác định sao cho mỗi byte chứa một số lẻ các số "1". Bởi vậy một sai sót đơn lẻ có thể phát hiện được trong mỗi nhóm 8 bit. Các bit kiểm tra bị bỏ qua trong quá trình tính toán bảng khoá.

Các bước để tính khóa:

1. Với một khoá K 64 bit cho trước, ta loại bỏ các bit kiểm tra tính chẵn lẻ bằng cách áp dụng hoán vị các bit của K theo phép hoán vị cố định PC-1, sau đó chia khóa thành 2 phần C_0: 28 bit đầu, D_0: 28 bit cuối.

$PC\text{-}1(K) = C_0D_0$

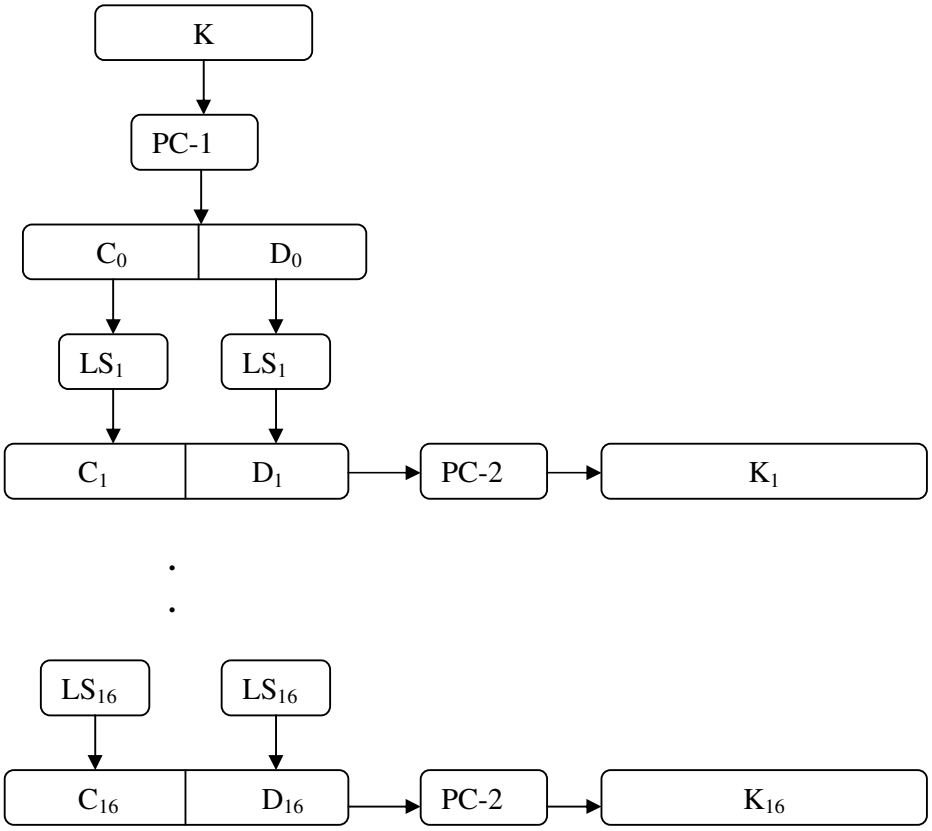
2. Với i thay đổi từ 1 đến 16: Thực hiện dịch trái 1 hoặc 2 bit phụ thuộc vào số vòng lặp.

3. Các bit của khóa được chọn ra theo hoán vị nén PC-2 (hoán vị lựa chọn): 56 bit -> 48 bit.

$C_i = LS_i(C_{i-1})$, $D_i = LS_i(Di\text{-}1)$

Vòng	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bít dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tính bảng khoá DES.



Tính hàm f

Hàm f có hai biến vào: biến thứ nhất A (thành phần R_{i-1}) là xâu bit độ dài 32, biến thứ hai J (khóa K_i) là một xâu bit độ dài 48. Đầu ra của f là một xâu bit độ dài 32.

Các bước sau được thực hiện:

1. A (nửa phải của dữ liệu R_{i-1} được mở rộng thành xâu bit độ dài 48 theo một hàm mở rộng cố định), E(A) gồm 32 bit được mở rộng thành 48 bit theo hoán vị mở rộng E (nhằm mục đích tạo ra dữ liệu có cùng kích cỡ với dòng khóa để thực hiện phép toán XOR).

2. Tính E(A)⊕J và viết kết quả thành một chuỗi 8 khối 6 bit $B_1B_2B_3B_4B_5B_6B_7B_8$.

3. Mỗi khối B_j sau đó được đưa vào một hàm S_j (S-box): $C_j=S_j(B_j)$ trả về một khối 4 bit.

Mỗi khối được thực hiện trên một hộp S riêng (B_1–S_1, …B_8–S_8,).

Hộp S là bảng gồm 4 hàng và 16 dòng, với khối bit có độ dài 6, kí hiệu $B_i = b_1b_2b_3b_4b_5b_6$.

Tính $S_j(B_j)$ như sau:

+ Hai bit b_1b_6 xác định biểu diễn nhị phân của hàng r của S_j ($0 \le r \le 3$) và bốn bit ($b_2b_3b_4b_5$) xác định biểu diễn nhị phân của cột c của S_j ($0 \le c \le 15$).

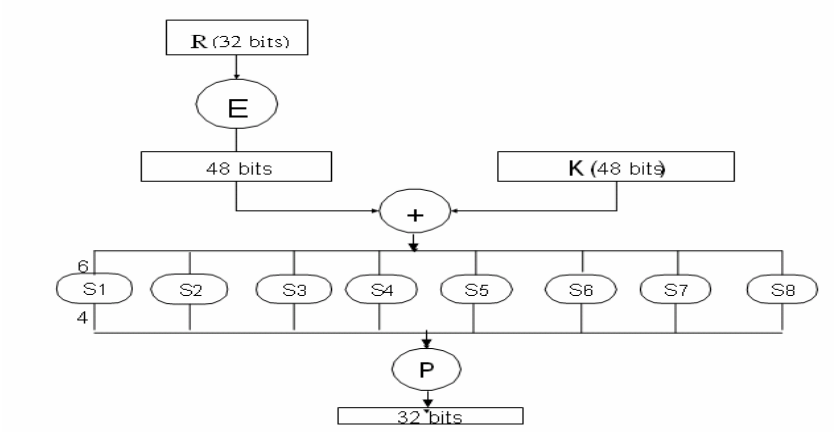
+ $S_j(B_j)=S_j(r,c)$; (r–hàng, c-cột) trong hộp S). Phần tử này viết dưới dạng nhị phân là một xâu bit có độ dài 4.

+ Bằng cách tương tự tính các $C_j = S_j(B_j)$, $1 \le j \le 8$.

4. Ghép các xâu bit $C = C_1C_2... C_8$ có độ dài 32 được hoán vị theo phép hoán vị cố định P, xâu kết quả là P(C) được xác định là f(A,J).

$$f(R_{i-1}, K_i) = P(S_1(B_1) \ldots. S_8(B_8))$$

Hàm f của DES



5.4 Một số hệ mã hoá công khai

5.4.1 Khái niệm chung

Trong phần này, chúng ta sẽ đề cập đến những phép biến đổi của mã khoá công khai hay mã không đối xứng, trong hệ mã khoá công khai, mỗi thực thể A có một khoá công khai e và tương ứng là một khoá riêng d. Hệ thống này bảo đảm việc tính d từ e là không thể làm được. Khoá công khai xác định một phép mã hoá Ee, trong khi đó khoá riêng xác định phép giải mã Dd. Bất kỳ thực thể B nào muốn gửi một văn bản tới A thu được một bản sao xác thực từ khoá công khai của A là e, sử dụng phép mã hoá để thu được bản mã c=Ee(m) và truyền tới A. Để giải mã c, A áp dụng phép giải mã, thu được văn bản gốc m=Dd(c).

Khoá công khai không cần giữ bí mật, một thuận lợi chính của những hệ thống này là cung cấp tính xác thực khoá công khai, thường dễ hơn so với việc bảo đảm phân phối khoá bí mật. Mục đích chính của mã khoá công khai là cung cấp sự bí mật và sự tin cậy.

Sơ đồ mã khoá công khai về cơ bản là chậm hơn so với thuật toán mã hoá đối xứng. Vì lý do này, mã khoá công khai được sử dụng phổ biến nhất trên thực tế cho việc truyền tải khoá, sau đó được sử dụng để mã hoá khối dữ liệu bằng thuật toán đối xứng, và những ứng dụng khác bao gồm sự toàn vẹn dữ liệu và sự xác định quyền. Một số tính chất quan trọng của mã khóa công khai

- Mã khóa công khai có tính chất bất đối xứng, sử dụng 2 khóa riêng biệt tương phản với mã hóa qui ước có tính đối xứng là chỉ sự dụng 1 khóa. Việc sử dụng 2 khóa có tầm quan trọng sâu sắc trong lĩnh vực cần tính bí mật, phân bố khóa và sự chứng thực quyền. Một khóa cho mã hóa (khóa công khai) và

một khóa khác (có quan hệ với khóa trên) cho giải mã (khóa bí mật, khóa riêng chỉ có người nhận mới biết mã này).

- Một đối tượng B muốn gửi tin cho A thì phải dùng khóa công khai của A để mã hóa thông tin, để giải mã được bản tin B gửi thì A sử dụng khóa riêng của mình để giải mã.

- Giải thuật mã hóa có đặc điểm là nếu biết giải thuật mã và khóa mã hóa (khoá công khai) tuy nhiên việc tính ra khóa bí mật hay khả năng giải mã là không khả thi.

Các bước cần thiết trong quá trình mã hóa công khai:

+ Người nhận bản mã tự tạo ra một cặp khóa để dùng cho mã hóa và giải mã đoạn tin mà mình sẽ nhận.

+ Công bố rộng rãi khóa mã hóa bằng cách đặt khóa vào một thanh ghi hay một file công khai. Đây là khóa công khai, khóa còn lại được giữ riêng.

+ Nếu A muốn gửi một đoạn tin tới B thì A mã hóa đoạn tin bằng khóa công khai của B. Khi B nhận đoạn tin mã hóa, giải mã bằng khóa bí mật của mình.

Một số hệ mã công khai thông dụng

5.4.2 Hệ mã RSA (R.Rivest, A.Shamir, L.Adleman)

Khái niệm hệ mật mã RSA đã được ra đời năm 1976 bởi các tác giả R.Rivets, A.Shamir, và L.Adleman. Hệ mã hoá này dựa trên cơ sở của hai bài toán :

+ Bài toán Logarithm rời rạc (Discrete logarith)

+ Bài toán phân tích thành thừa số nguyên tố

Sơ đồ mã hoá RSA là sơ đồ mã hoá khối, đoạn tin được mã hoá từng khối với mỗi khối có giá trị < n với n là số nguyên đủ lớn.

Hệ mã RSA là hệ mã dựa vào bài toán logarithm rời rạc và bài toán phân tích một số nguyên thành tích các thừa số nguyên tố, là hệ mã được sử dụng

rộng rãi nhất. Nó cung cấp cả sự bí mật và chữ ký điện tử, và tính bảo mật của nó là cơ sở cho độ khó trong vấn đề tìm thừa số nguyên tố.

Thuật toán

Tạo khoá:

Mỗi thực thể tạo một khoá công khai và một khoá riêng tương ứng.

Thực thể A cần làm công việc sau:

1. Tạo 2 số nguyên tố lớn p và q bất kỳ có cỡ xấp xỉ nhau.
2. Tính n = p×q và $\Phi(n)$= (p-1)(q-1).
3. Chọn 1 số nguyên e bất kỳ, 1<e<$\Phi(n)$ sao cho UCLN(e, $\Phi(n)$) = 1.
4. Sử dụng thuật toán Euclid mở rộng để tính d, 1<d<$\Phi(n)$ sao cho: e×d ≡ 1 (mod $\Phi(n)$)
5. Khoá công khai của A là (n,e), khoá riêng của A là d.

Mã hoá: B mã hoá một văn bản gửi cho A.

1. Nhận được khoá xác thực công khai của A là (n,e).
2. Trình bày văn bản như một số nguyên m thuộc [0, n-1].
3. Tính $c = m^e \bmod n$.
4. Gửi bản mã c cho A.

Giải mã: Để khôi phục bản rõ m từ bản mã c, A phải thực hiện công việc sau: Sử dụng khoá riêng d để tính $m = c^d \bmod n$.

Chú ý: Số λ=BCNN(p-1, q-1) còn được gọi là số mũ tự nhiên của n có thể được sử dụng để thay cho $\Phi(n)$= (p-1)(q-1) trong việc tạo khoá RSA. Để ý rằng λ là một ước số đúng của $\Phi(n)$. Sử dụng λ có thể tính toán với giải mã số mũ nhỏ hơn và kết quả là việc giải mã sẽ nhanh hơn.

Ví dụ

Tạo khoá: A chọn p = 2357, q = 2551.

+ Tính n = p×q = 6012707, $\Phi(n)$ = (p-1)(q-1) = 6007800.

+ A chọn e = 3674911.

+ Sử dụng thuật toán Euclid mở rộng tính được d = 422191, sao cho $e \times d \equiv 1 (\mathrm{mod}\ \Phi(n))$

Khoá công khai của A là (n = 6012707, e = 3674911), khoá riêng của A là d = 422191.

Mã hoá: Để mã hoá m = 5234673, B sử dụng thuật toán tính luỹ thừa nhanh để tính:

$c = m^e \bmod n = 5234673^{3674911} \bmod 6012707 = 3650502$

Giải mã: Để giải mã c, A thực hiện tính:

$c^d \bmod n = 3650502^{422191} \bmod 6012707 = 5234673$

Chú ý: Có nhiều cách để tăng tốc độ mã hoá và giải mã RSA nhưng với một số cải tiến thì tốc độ mã hoá/giải mã RSA về căn bản vẫn chậm hơn so với các thuật toán mã hoá đối xứng. Trong thực tế, hệ mã RSA được sử dụng phổ biến để truyền khoá đã được tạo bằng thuật toán đối xứng và mã hoá những mục dữ liệu nhỏ.

Lựa chọn số nguyên tố.

Số nguyên tố p và q nên được chọn sao cho việc phân tích n = p×q là không thể tính toán được, p và q nên có cùng cỡ và phải đủ lớn. Một chú ý khác là với số nguyên tố p và q thì hiệu p - q không nên quá nhỏ vì nếu p-q nhỏ thì $p \approx q$ và do vậy $p \approx \sqrt{n}$. Khi đó có thể dễ dàng phân tích n thành thừa số bằng cách chia thử cho tất cả các số nguyên lẻ cho tới $\sqrt{n}$.

Một số tính chất của hệ RSA

Trong các hệ mật mã RSA, một bản tin có thể được mã hoá trong thời gian tuyến tính. Đối với các bản tin dài, độ dài của các số được dùng cho các khoá có thể được coi như là hằng. Tương tự như vậy, nâng một số lên luỹ thừa

được thực hiện trong thời gian hằng, các số không được phép dài hơn một độ dài hằng. Thực ra tham số này che dấu nhiều chi tiết cài đặt có liên quan đến việc tính toán với các con số dài, chi phí của các phép toán thực sự là một yếu tố ngăn cản sự phổ biến ứng dụng của phương pháp này. Phần quan trọng nhất của việc tính toán có liên quan đến việc mã hoá bản tin. Nhưng chắc chắn là sẽ không có hệ mã hoá nào hết nếu không tính ra được các khoá của chúng là các số lớn. Các khoá cho hệ mã hoá RSA có thể được tạo ra mà không phải tính toán quá nhiều.

Mỗi số nguyên tố lớn có thể được phát sinh bằng cách đầu tiên tạo ra một số ngẫu nhiên lớn, sau đó kiểm tra các số kế tiếp cho tới khi tìm được một số nguyên tố. Các bước tính p dựa vào thuật toán Euclid.

Như phần trên đã trình bày trong hệ mã hoá công khai thì khoá giải mã (private key) kB và các thừa số p,q là được giữ bí mật và sự thành công của phương pháp là tuỳ thuộc vào kẻ địch có khả năng tìm ra được giá trị của kB hay không nếu cho trước N và KB. Rất khó có thể tìm ra được kB từ KB vì cần phải xác định được p và q. Như vậy cần phân tích N ra thành thừa số để tính p và q, nhưng việc phân tích ra thừa số là một việc làm tốn rất nhiều thời gian, với kỹ thuật hiện đại ngày nay thì cần tới hàng triệu năm để phân tích một số có 200 chữ số ra thừa số.

Độ an toàn của thuật toán RSA dựa trên cơ sở những khó khăn của việc xác định các thừa số nguyên tố của một số lớn.

5.4.3 Hệ mã Rabin

Đây là hệ mật có độ an toàn cao về mặt tính toán chống lại được cách tấn công bản rõ

Thuật toán

Tạo khoá: Mỗi thực thể tạo một khoá công khai và một khoá riêng tương ứng. Mỗi thực thể A cần thực hiện các công việc sau:

1. Tạo hai số nguyên tố lớn p và q có cỡ xấp xỉ nhau.
2. Tính n=p×q.
3. Khoá công khai của A là n, khoá riêng của A là (p,q).

Mã hoá: B mã hoá văn bản m gửi cho A.

1. Nhận được khoá công khai xác thực của A là n.
2. Biểu diễn đoạn văn bản là một số nguyên trong miền: {0, 1, . . , n-1}
3. Tính $c = m^2 \bmod n$.
4. Gửi bản mã c cho A.

Giải mã: Khôi phục bản rõ m từ c, cần phải dùng thuật toán tìm căn bậc hai theo modulo n từ số nguyên tố p và q đã cho để tính ra 4 căn bậc hai m_1, m_2, m_3 và m_4 của c theo modulo n. Khi đó văn bản đã gửi là một trong các giá trị m_1, m_2, m_3 hoặc m_4. A bằng cách này hay cách khác quyết định đâu là m.

Chú ý : Giả sử p và q đều được chọn sao cho $\equiv 3 \pmod 4$ thì việc tính toán 4 căn bậc 2 của c theo modulo n được thực hiện đơn giản theo thuật toán sau:

+ Sử dụng thuật toán Euclid mở rộng để tìm số nguyên a và b thoả mãn $a \times p + b \times q = 1$. Chú ý rằng a và b có thể được tính chỉ 1 lần trong suốt quá trình tạo khoá.
+ Tính $r = c^{(p+1)/4} \bmod p$.
+ Tính $s = c^{(q+1)/4} \bmod q$.
+ Tính $x = (aps + bqr) \bmod n$.
+ Tính $y = (aps - bqr) \bmod n$.
+ Bốn căn bậc 2 của c theo modulo n lần lượt là x, -x mod n, y và -y mod n.

Ví dụ

Tạo khoá: Chọn số nguyên tố p = 277, q = 331, tính n = p×q = 91687. Khoá công khai của A là n = 91687, khoá riêng là (p = 277, q= 331).

Mã hoá: Giả sử rằng 6 bit cuối cùng của văn bản gốc là quy định để tái tạo trước khi mã hoá. Yêu cầu mã hoá 10 bit văn bản $\overline{m}$ = 1001111001, B tái tạo lại 6 bit cuối cùng của $\overline{m}$ để thu được 16 bit văn bản m = 1001111001111001, biểu diễn ở dạng thập phân là m = 40569 sau đó B tính $c = m^2 \bmod n = 40569^2 \bmod 91687 = 62111$.

Giải mã: Để giải mã c, A dùng thuật toán tính được 4 căn bậc hai của c mod n là m_1 = 69654, m_2 = 22033, m_3 = 40569, m_4 = 51118 biểu diễn ở dạng nhị phân là:

m_1 = 10001000000010110, m_2 = 1010110000010001

m_3 = 1001111001111001, m_4 = 1100011110101110

Ta thấy chỉ m_3 có phần mở rộng quy định, khi đó A tiến hành giải mã c (hay m_3) để khôi phục văn bản gốc $\overline{m}$ = 1001111001.

Tính bảo mật của hệ mã khoá công khai Rabin

Do việc phân tích n thành thừa số là rất khó nên lược đồ mã khoá công khai Rabin là minh chứng bảo đảm chống lại sự tấn công của đối phương. Tuy nhiên, lược đồ mã khoá công khai Rabin không chống lại được sự tấn công bản mã chọn trước. Mặt hạn chế của lược đồ mã khoá công khai Rabin là người nhận phải lựa chọn chính xác bản rõ từ 4 khả năng. Sự nhập nhằng trong giải mã có thể dễ dàng khắc phục trong thực tế bằng cách thêm phần mở rộng vào bản rõ trước khi mã hoá. Khi đó, với xác suất cao, chính xác một trong bốn căn bậc 2: m_1, m_2, m_3, m_4 của bản mã c sẽ có phần mở rộng này, và người nhận sẽ chọn đó là bản rõ mong đợi. Nếu không có căn bậc 2 nào của c có phần mở rộng này thì người nhận từ chối. Nếu phần mở rộng được sử dụng

như đã nói ở trên, lược đồ Rabin không dễ bị tấn công bằng bản mã chọn trước.

5.4.4 Hệ mã Elgamal

Hệ mã Elgamal xây dựng dựa trên bài toán logarithm rời rạc là bài toán được dùng nhiều trong thủ tục mật mã.

Chúng ta sẽ bắt đầu bằng việc mô tả bài toán khi thiết lập trường hữu hạn Z_p trong đó p là số nguyên tố. Nhớ lại rằng nhóm nhân Z^*_P là nhóm cyclic và phần tử sinh của Z^*_P được gọi là phần tử nguyên thuỷ. Bài toán logarithm rời rạc trong Z_p là đối tượng trong nhiều công trình nghiên cứu và được xem là bài toán khó. Không có một thuật toán thời gian đa thức nào cho bài toán logarithm rời rạc, để gây khó khăn cho các phương pháp tấn công đã biết, p phải có ít nhất 150 chữ số và (p-1) phải có ít nhất một thừa số nguyên tố lớn. Lợi thế của bài toán logarithm rời rạc trong xây dựng hệ mã là khó tìm được các logarithm rời rạc, song bài toán ngược lấy luỹ thừa lại có thể tính toán hiệu quả theo thuật toán "bình phương và nhân". Nói cách khác, luỹ thừa theo modulo p là hàm một chiều với các số nguyên tố p thích hợp.

Thuật toán

Tạo khoá: Mỗi thực thể tạo một khoá công khai và một khoá riêng tương ứng.

1. Tạo một số nguyên tố p và phần tử sinh $\alpha \in$ nhóm nhân Z^*_P.
2. Chọn một số nguyên a, $1 \leq a \leq p-2$, và tính $\alpha^a \bmod p$.
3. Khoá công khai của A là (p, α, α^a); khoá riêng là a.

Mã hoá: B mã hoá văn bản m gửi cho A.

1. Thu được khoá công khai của A (p, α, α^a).
2. Biểu diễn văn bản như một số nguyên m trong {0,1,...,p-1}.
3. Chọn ngẫu nhiên một số nguyên k, $1 \leq k \leq p-2$.

4. Tính $\gamma = \alpha^k$ mod p và $\delta = m \times (\alpha^a)^k$ mod p.

5. Gửi bản mã c = (γ, δ) cho A.

Giải mã: Khôi phục bản rõ m từ c bằng việc tính $(\gamma - 1) \times \delta$ mod p.

Ví dụ

Tạo khoá: Chọn số nguyên tố p=2357 và phần tử sinh α=2$\in$ Z^*_{2357}. A chọn khoá riêng a = 1751 và tính α^a mod p = 2^{1751} mod 2357 = 1185. Khoá công khai của A là (p = 2357, α = 2, α^a = 1185).

Mã hoá: Để mã hoá một văn bản m=2035, B chọn ngẫu nhiên một số nguyên k = 1520 và tính γ=2^{1520} mod 2357 = 1430,

δ = 2035×1185^{1520} mod 2357 = 697**.** B gửi γ=1430 và δ=697 cho A.

Giải mã: Để giải mã, A thực hiện tính: m = 872*697 mod 2357 = 2035.

Chú ý: Một bất lợi của mã hoá Elgamal là sự mở rộng của văn bản, bản mã có thể dài gấp 2 lần so với bản rõ. Mã hoá Elgamal là một trong những lược đồ mã hoá sử dụng sự ngẫu nhiên trong tiến trình mã hoá. Những nguyên tắc cơ bản đằng sau kỹ thuật mã hoá ngẫu nhiên là sử dụng tính ngẫu nhiên để tăng thêm sự bảo mật bằng mật mã của tiến trình mã hoá theo một trong các phương thức sau:

1. Tăng dần khoảng trống trong bản rõ một cách phù hợp.
2. Ngăn ngừa hoặc làm suy giảm sự có hiệu lực của sự tấn công bản rõ chọn trước thông qua ánh xạ một - nhiều từ bản rõ đến bản mã.

Tính bảo mật của mã hoá Elgamal

Vấn đề bẻ khoá lược đồ mã hoá Elgamal, khôi phục m từ p, α, α^a, γ, và δ tương đương với giải quyết vấn đề Diff-Hellman. Trên thực tế, lược đồ mã hoá Elgamal được xem như sự thay đổi khoá Diff-Hellman để quyết định khoá α^{ak} , và việc mã hoá văn bản bằng tính nhân với khoá đó. Vì lý do này,

tính bảo mật của lược đồ mã hoá Elgamal được gọi là cơ sở trong vấn đề logarithm rời rạc trong Z^*_P .

5.4.5 Hệ mã MHK (Merkle -Hellman Knapsack)

Sơ đồ mã khoá công khai ba lô dựa trên cơ sở của bài toán tập con, quan điểm cơ bản là chọn một trường hợp của bài toán tổng con mà dễ dàng tìm lời giải, sau đó che giấu nó như một trường hợp của bài toán tổng con tổng quát khó có hy vọng giải được. Khoá được thiết lập ban đầu có thể dùng như khoá riêng, còn khoá được thiết lập sau khi biến đổi là khoá công khai. Sơ đồ mã hoá MHK che giấu lời giải bằng phép nhân theo modulo và phép hoán vị.

Định nghĩa một dãy siêu tăng là một dãy các số nguyên dương (b_1,b_2, ..., b_n) thoả mãn tính chất $b_i > \sum_{j=1}^{i-1} b_j$ với mỗi i, $2 \le i \le n$.

Thuật toán

Tạo khoá: Mỗi thực thể tạo một khoá công khai và một khoá riêng tương ứng. Một số nguyên n được cố định như một tham số hệ thống phổ biến.

1. Chọn một dãy siêu tăng (b_1,b_2, ..., b_n) và modulo M sao cho $M > b_1 + b_2 + ... + bn$.
2. Chọn số nguyên ngẫu nhiên W $1 \le W \le M-1$ sao cho UCLN(W, M) = 1.
3. Chọn một phép hoán vị ngẫu nhiên π của n số nguyên {1,2,...,n}.
4. Tính $a_i = Wb_{\pi(i)} \bmod M$ với i = 1, 2, ..., n.
5. Khoá công khai là (a_1, a_2, ..., a_n), khoá riêng là (π, M, W, (b_1,b_2, ..., b_n)).

Mã hoá: B mã hoá văn bản m gửi cho A.

1. Thu được khoá công khai (a_1, a_2, ..., an) của A.
2. Biểu diễn văn bản m như một chuỗi nhị phân có độ dài n, $m=m_1m_2...m_n$.
3. Tính số nguyên $c = m_1a_1 + m_2a_2 + ... + m_na_n$.

4. Gửi bản mã c cho A.

Giải mã: Để khôi phục bản rõ m từ c, cần thực hiện các công việc sau:

1. Tính $d = W^{-1}c \mod M$.
2. Sử dụng thuật toán tìm lời giải bài toán tổng dãy siêu tăng để tìm các số nguyên $r_1,r_2,...,r_n, r_i \in \{0,1\}$, sao cho $d=r_1b_1+r_2b_2+ ... +r_nb_n$.
3. Các bit văn bản m là $m_i = r_{\pi(i)}$, i = 1, 2, ..., n.

Ví dụ

Tạo khoá: Lấy n=6, chọn dãy siêu tăng (12, 17, 33, 74, 157, 316), M=737, W=635 và phép hoán vị π của (1, 2, 3, 4, 5, 6) được định nghĩa bởi $\pi(1)=3$, $\pi(2)=6$, $\pi(3)=1$, $\pi(4)=2$, $\pi(5)=5$, $\pi(6)=4$.

Khoá công khai là (319, 196, 250, 477, 200, 559), Khoá riêng là (π, M, W, (12, 17, 33, 74, 157, 316)).

Mã hoá: Văn bản m = 101101, tính: c = 319 + 250 + 477 + 559 = 1605

Giải mã: Tính $d = W^{-1}c \mod M = 136$, và tìm lời giải cho bài toán tổng dãy siêu tăng: $136 = 12r_1 + 17r_2 + 33r_3 + 74r_4 + 157r_5 + 316r_6$, nhận được 136 = 12 + 17 + 33 + 74. Từ đó $r_1 = 1$, $r_2 = 1$, $r_3 = 1$, $r_4 = 1$, $r_5 = 0$, $r_6 = 0$. Sử dụng phép hoán vị π được các bit văn bản m: $m_1 = r_3 = 1$, $m_2 = r_6 = 0$, $m_3 = r_1 = 1$, $m_4 = r_2 = 1$, $m_5 = r_5 = 0$, $m_6 = r_4 = 1$.

Tính bảo mật của mã hoá MHK

Lược đồ mã hoá MHK có thể bị bẻ khoá bởi thuật toán thời gian đa thức. Trong cách thiết lập khoá công khai, thuật toán này tìm một cặp số nguyên U', M' sao cho U'/M' là tỉ lệ với U/M (W và M là một phần khoá riêng, và $U=W^{-1} \mod M$) và sao cho các số nguyên $b'_i = U'a_i \mod M$, $1 \le i \le n$ từ dãy siêu tăng. Dãy này có thể được một đối thủ sử dụng thay thế vào dãy $(b_1,b_2, ..., b_n)$ để giải mã văn bản.

TÀI LIỆU THAM KHẢO

[1] Đặng Văn Chuyết, Nguyễn Tuấn Anh, "Cơ sở lý thuyết truyền tin" – Tập 1 và 2, Nhà xuất bản Giáo dục, 1998.

[2] Robert B.Ash, "Information theory ", Nhà xuất bản Dover, inc, 1990.

[3] Masud Mansuripur, "Introduction to information theory ", Nhà xuất bản Prentice-Hall, Inc, 1987.

[4] I. Cziszar and J. Korner. Information theory, "Coding Theorems for Discrete Memoryless Systems". Academic Press, New York. 1997. 2nd edition.

[5] D. Hammer, A. Romashenko, A. Shen, N. Vereshchagin, "Inequalities for Shannon entropies and Kolmogorov complexities", Proceedings of CCC'97 Conference, Ulm. Final version: Inequalities for Shannon entropy and Kolmogorov Complexity, Journal of Computer and System Sciences, v. 60, p. 442{464 (2000).

[6] A. Shen, "Algorithmic Information Theory and Kolmogorov Complexity", Lecture notes of an introductory course. Uppsala University Technical Report 2000-034. Available online at
http://www.it.uu.se/research/publications/reports/2000-034

[7] Bar-Yam, Yaneer, "Dynamics of Complex Systems (Studies in Nonlinearity)", Westview Press, Boulder, 1997.

[8] Campbell, Jeremy, Grammatical Man, "Information, Entropy, Language, and Life, Simon and Schuster", New York, 1982.

[9] Cover, T. M., and Thomas J. A., "Elements of Information Theory", John Wiley and Sons, New York, 1991.

[10] Gatlin, L. L., "Information Theory and the Living System", Columbia University Press, New York, 1972.

[11] Hamming, R. W., “Coding and information theory”, 2nd ed, Prentice-Hall, Englewood Cliffs, 1986.

[12] Landauer, R., “The physical nature of information”, Phys. Lett. A, 217 188, 1996.

MỤC LỤC

LỜI MỞ ĐẦU 1

CHƯƠNG 1. NHỮNG KHÁI NIỆM CƠ BẢN 3

1.1 Giới thiệu về lý thuyết thông tin 3
1.2 Hệ thống truyền tin 3
1.2.1 Các quan điểm để phân loại các hệ thống truyền tin 4
1.2.2 Sơ đồ truyền tin và một số khái niệm trong hệ thống truyền tin..... 4
1.3 Nguồn tin nguyên thuỷ..... 5
1.3.1 Khái niệm chung 5
1.3.2 Bản chất của thông tin theo quan điểm truyền tin 7
1.4 Hệ thống kênh tin..... 10
1.4.1 Khái niệm..... 10
1.4.2 Phân loại môi trường truyền tin 11
1.4.3 Mô tả sự truyền tin qua kênh: 11
1.5 Hệ thống nhận tin 13
1.6 Một số vấn đề cơ bản của hệ thống nhận tin..... 13
1.6.1 Hiệu suất..... 13
1.6.2 Độ chính xác..... 13
1.7 Rời rạc hóa một nguồn tin liên tục..... 13
1.7.1 Quá trình lấy mẫu..... 14
1.7.2 Quá trình lượng tử hóa..... 16
1.8 Điều chế và giải điều chế 17
1.8.2 Giải điều chế 18

CHƯƠNG 2. TÍN HIỆU 19

2.1 Một số khái niệm cơ bản..... 19
2.1.1 Tín hiệu duy trì: 19
2.1.2 Tín hiệu xung 19
2.2 Phân tích phổ cho tín hiệu..... 20
2.2.2 Tích phân Fourier và phổ liên tục 27
2.2.3 Phổ các tín hiệu điều chế 28
2.3 Nhiễu trắng..... 33

CHƯƠNG 3. LƯỢNG TIN, ENTROPI NGUỒN RỜI RẠC 35

3.1 Độ đo thông tin 35
3.1.2 Độ đo thông tin. 35
3.2 Lượng tin của nguồn rời rạc..... 37

3.2.1 Mối liên hệ của lượng tin và lý thuyết xác suất 37
3.2.3 Tính chất của lượng tin 45
3.2.4 Lượng tin trung bình 46
3.3 Entropi của nguồn rời rạc 47
3.3.1 Khái niệm entropi 47
3.3.2 Tính chất của entropi 47
3.3.3 Entropi đồng thời và Entropi có điều kiện 48
3.3.4 Entropi nguồn Markov 49
3.4 Mối quan hệ giữa lượng tin tương hỗ trung bình và Entropi 50
3.5 Tốc độ lập tin nguồn rời rạc và thông lượng kênh rời rạc 52
3.5.1 Tốc độ lập tin 52
3.5.2 Thông lượng kênh 54
CHƯƠNG 4. LÝ THUYẾT MÃ 56
4.1 Khái niệm mã và điều kiện thiết lập mã 56
4.1.1 Mã hiệu và các thông số cơ bản 56
4.1.2 Điều kiện thiết lập bộ mã 58
4.2 Các phương pháp biểu diễn mã 60
4.2.1 Biểu diễn bằng bảng liệt kê (Bảng đối chiếu mã) 60
4.2.2 Biểu diễn bằng toạ độ mã 60
4.2.3 Đồ hình mã 61
4.2.4 Phương pháp hàm cấu trúc mã 62
4.3 Mã có tính phân tách được, mã có tính prefix 62
4.3.1 Điều kiện để mã phân tách được 63
4.3.2 Mã có tính prefix 65
4.3.3 Bất đẳng thức Kraft 66
4.4 Mã thống kê tối ưu 67
4.4.1 Giới hạn độ dài trung bình của từ mã 67
4.4.2 Tiêu chuẩn mã thống kê tối ưu 68
4.4.3 Mã thống kê Fano –Shanon 69
4.5 Thuật toán mã hoá Lempel-Ziv 83
4.6 Mã chống nhiễu 85
4.6.1 Khái niệm về mã phát hiện sai và sửa sai 86
4.6.2 Cơ chế phát hiện sai 87
4.6.3 Xây dựng bộ mã sửa sai bằng bảng chọn 89
4.6.4 Xây dựng bộ mã sửa sai bằng trọng số Hamming và khoảng cách Hamming 90
4.5.5 Một số biện pháp xây dựng bộ mã phát hiện sai và sửa sai 92
4.7 Mã tuyến tính 94
4.7.2 Nguyên lý giải mã 96

4.7.3 Một số giới hạn của mã tuyến tính....98
4.8 Mã vòng....99
4.8.1 Khái niệm:....99
4.8.2 Nguyên lý lập mã....100
4.8.3 Nguyên lý giải mã....102
CHƯƠNG 5. GIỚI THIỆU VỀ HỆ MẬT MÃ 105
5.1 Khái niệm hệ mật mã....105
5.2 Phân loại các hệ thống mật mã....105
5.3 Hệ mã cổ điển (Symmetric-key encryption)....106
5.3.1 Khái niệm....106
5.3.2 Một số hệ mã cổ điển....106
5.3.3 Hệ mã DES....117
5.4 Một số hệ mã hoá công khai....121
5.4.1 Khái niệm chung....121
5.4.2 Hệ mã RSA....123
5.4.3 Hệ mã Rabin....126
5.4.4 Hệ mã Elgamal....129
5.4.5 Hệ mã MHK....130
TÀI LIỆU THAM KHẢO....131
MỤC LỤC....133